SIEMENS

SIMATIC

S7
Система автоматизации S7-1200

Системное руководство

Предисловие

Обзор продукта 1

Новые функции 2

ПО для программирования STEP 7 3

Монтаж 4

Основы PLC 5

Конфигурация устройства 6

Основы программирования 7

Инструкции 8

Расширенные инструкции 9

Технологические инструкции 10

Коммуникация 11

Веб-сервер 12

Коммуникационный процессор и Modbus TCP 13

Коммуникации TeleService (SMTP email) 14

Онлайн инструменты и диагностика 15

Технические данные A

Расчет баланса мощностей B

Информация для оформления заказа C

Замена устройств и совместимость по запасным частям D

V4.4 11/2019
A5E50581466-AN

Правовая справочная информация

Система предупреждений

Данная инструкция содержит указания, которые Вы должны соблюдать для Вашей личной безопасности и для предотвращения материального ущерба. Указания по Вашей личной безопасности выделены предупреждающим треугольником, общие указания по предотвращению материального ущерба не имеют этого треугольника. В зависимости от степени опасности, предупреждающие указания представляются в убывающей последовательности следующим образом:

⚠ ОПАСНО
означает, что непринятие соответствующих мер предосторожности **приводит** к смерти или получению тяжелых телесных повреждений.

⚠ ПРЕДУПРЕЖДЕНИЕ
означает, что непринятие соответствующих мер предосторожности **может** привести к смерти или получению тяжелых телесных повреждений.

⚠ ОСТОРОЖНО
означает, что непринятие соответствующих мер предосторожности может привести к получению незначительных телесных повреждений.

ВНИМАНИЕ
означает, что непринятие соответствующих мер предосторожности может привести к материальному ущербу.

При возникновении нескольких степеней опасности всегда используется предупреждающее указание, относящееся к наивысшей степени. Если в предупреждении с предупреждающим треугольником речь идет о предупреждении ущерба, причиняемому людям, то в этом же предупреждении дополнительно могут иметься указания о предупреждении материального ущерба.

Квалифицированный персонал

Работать с изделием или системой, описываемой в данной документации, должен только **квалифицированный персонал**, допущенный для выполнения поставленных задач и соблюдающий соответствующие указания документации, в частности, указания и предупреждения по технике безопасности. Квалифицированный персонал в силу своих знаний и опыта в состоянии распознать риски при обращении с данными изделиями или системами и избежать возникающих угроз.

Использование изделий Siemens по назначению

Соблюдайте следующее:

⚠ ПРЕДУПРЕЖДЕНИЕ
Изделия Siemens разрешается использовать только для целей, указанных в каталоге и в соответствующей технической документации. Если предполагается использовать изделия и компоненты других производителей, то обязательным является получение рекомендации и/или разрешения на это от фирмы Siemens. Исходными условиями для безупречной и надежной работы изделий являются надлежащая транспортировка, хранение, размещение, монтаж, оснащение, ввод в эксплуатацию, обслуживание и поддержание в исправном состоянии. Необходимо соблюдать допустимые условия окружающей среды. Обязательно учитывайте указания в соответствующей документации.

Товарные знаки

Все наименования, обозначенные символом защищенных авторских прав ®, являются зарегистрированными товарными знаками компании Siemens AG. Другие наименования в данной документации могут быть товарные знаки, использование которых третьими лицами для их целей могут нарушать права владельцев.

Исключение ответственности

Мы проверили содержимое документации на соответствие с описанным аппаратным и программным обеспечением. Тем не менее, отклонения не могут быть исключены, в связи с чем мы не гарантируем полное соответствие. Данные в этой документации регулярно проверяются и соответствующие корректуры вносятся в последующие издания.

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
ГЕРМАНИЯ

A5E50581466-AN
Ⓟ 11/2020 Возможны изменения

Copyright © Siemens AG 2019.
Все права защищены

Предисловие

Цель справочника

Семейство S7-1200 включает в себя различные программируемые логические контроллеры (PLC) для решения широкого спектра задач в области автоматизации. Благодаря компактной конструкции, наличию расширений, доступной цене и широкому набору команд, контроллеры S71200 идеально подходят для использования в различных приложениях. Модификации S7-1200 и различные возможности программирования STEP 7 (Страница 37) в Windows обеспечивают чрезвычайно высокий уровень гибкости при реализации решений автоматизации.

Настоящий справочник содержит информацию по установке и программированию контроллеров S7-1200 и предназначен для инженеров, программистов и обслуживающего персонала с базовым уровнем знаний в области систем автоматизации.

Необходимые базовые знания

Для работы с данным справочником необходимо иметь базовый объем знаний в области автоматизации и программируемых логических контроллеров.

Объем справочника

Справочник содержит описание следующих продуктов:

- STEP 7 Basic и Professional (Страница 37)
- S7-1200 CPU версия прошивки V4.4

Полный список продуктов S71200 можно найти в Технических данных (Страница 1433).

Сертификация, CE-маркировка, C-Tick и другие разрешения

Подробную информацию можно найти в Технических данных (Страница 1433).

Сервисная и техническая поддержка

Кроме документации Siemens предлагает через Интернете различные ноу-хау на сайте клиентской поддержки (http://support.industry.siemens.com).

Для решения технических вопросов, прохождения обучения или заказа продуктов S7, просьба обращаться в региональное представительство Siemens. Технически-обученные менеджеры, благодаря специальным знаниями в области использования, технологии и параметров различных продуктов Siemens, могут оказать быструю и эффективную помощью при возникновении проблем.

Документация и информация

Для S7-1200 и STEP 7 доступен большой объем документации и иной технической информации.

- Справочник по системе "Система автоматизации S7-1200" предлагает специальную информацию о принципе работы, программированию и техническим данным всей линейки продуктов S7-1200.
  Справочник по системе доступен в электронной форме (в формате PDF). Он и другие электронные справочники/руководства могут быть загружены или показаны через сайт Siemens Industry Online-Support (http://support.industry.siemens.com). Справочник по системе также доступен на CD с документацией, входящем в объем поставки каждого S7-1200 CPU.
- Интерактивная информационная система STEP 7 обеспечивает прямой доступ к концептуальной информации и специальным инструкцию по использованию и функциям пакета программирования, а также по базовым принципам работы процессоров SIMATIC CPU.
- Веб-сайт Siemens Industry Online-Support (http://support.industry.siemens.com) открывает доступ к электронным версиям (PDF) документации SIMATIC, включающей в себя и справочник по системе / информационную систему STEP 7. Данный документ можно скачать на странице документации на изделие. Такой доступ к интерактивной документации позволяет самостоятельно комбинировать темы из разных документов и путем "перетаскивания" создавать собственные справочники и руководства. Обновления ранее опубликованных справочников по системе также доступны на сайте Siemens Industry Online-Support.
  Для вызова интерактивной документации используется ссылка "mySupport" в левой части страницы. После выбрать опцию "Документация". Функция документация по ссылке mySupport доступна только для зарегистрированных пользователей.
- На веб-сайте Siemens Industry Online-Support также находится FAQ и другие полезные документы для S7-1200 и STEP 7.
- На форуме технической поддержки (https://support.industry.siemens.com/tf/ww/en/?Language=en&siteid=csius&treeLang=en&groupid=4000002&extranet=standard&viewreg=WW&nodeid0=34612486) можно следить или участвовать в дискуссиях по различным продуктам. Следующие форумы позволяют связываться с различными экспертами по продуктам.
  - Форум по S7-1200 (https://support.industry.siemens.com/tf/ww/en/threads/237?title=simatic-s7-1200&skip=0&take=10&orderBy=LastPostDate+desc)
  - Форум по STEP 7 Basic (https://support.industry.siemens.com/tf/ww/en/threads/243?title=step-7-tia-portal&skip=0&take=10&orderBy=LastPostDate+desc)

Указания по безопасности

Siemens предоставляет продукты и решения для обеспечения промышленной безопасности при эксплуатации производственных комплексов, систем, рабочих станций и сетей.

Для защиты производственных комплексов, систем, машинного оборудования и сетей от киберугроз необходимо внедрение и поддержка комплексной высокотехнологичной модели промышленной безопасности. Продукты и решения Siemens являются только одним из компонентов такой модели.

За предотвращение несанкционированного доступа к производственным комплексам, системам, рабочим станциям и сетям клиента несет ответственность клиент. Доступ систем, рабочих станций и их компонентов к корпоративной сети или сети Интернет должен быть организован только в необходимой степени и с применением соответствующих локальных мер безопасности (например, использование брандмауэров и деление сети на подсети).

Для получения дополнительных сведений о возможных мерах промышленной безопасности см. (https://www.siemens.com/industrialsecurity).

Продукты и решения Siemens постоянно совершенствуются для обеспечения максимальной степени безопасности. Siemens настоятельно рекомендует выполнять обновления сразу после их выпуска и всегда использовать самые последние версии продуктов. Использование неподдерживаемых версий продуктов и неприменение последних обновлений повышает риск киберугроз для клиента.

Для получения сведений об обновлениях продуктов, подпишитесь на RSS-канал Siemens по промышленной безопасности: (https://www.siemens.com/industrialsecurity).

Оглавление

Предисловие ... **3**

1 Обзор продукта ... **27**

1.1 Знакомство с контроллером S7-1200 ... 27

1.2 Возможности расширения CPU ... 31

1.3 Базовые панели HMI ... 32

2 Новые функции ... **35**

3 ПО для программирования STEP 7 ... **37**

3.1 Системные требования ... 37

3.2 Простая работа с различными представлениями ... 39

3.3 Эргономичные инструменты ... 40
3.3.1 Вставка инструкций в программе пользователя ... 40
3.3.2 Доступ к инструкциям на панели инструментов "Избранное" ... 41
3.3.3 Создание сложного уравнения с помощью простой инструкции ... 41
3.3.4 Добавление входов или выходов в LAD или FBD инструкцию ... 43
3.3.5 Расширяемые инструкции ... 44
3.3.6 Выбор версии для инструкции ... 44
3.3.7 Изменение внешнего вида и конфигурации STEP 7 ... 45
3.3.8 Перетаскивание между редакторами ... 46
3.3.9 Изменение рабочего состояния CPU ... 46
3.3.10 Изменение типа вызова DB ... 48
3.3.11 Временное отключение устройств от сети ... 48
3.3.12 Виртуальное исключение устройств из конфигурации ... 50

3.4 Совместимость с предыдущими версиями ... 50

4 Монтаж ... **53**

4.1 Руководства по монтажу устройств S71200 ... 53

4.2 Баланс мощности ... 55

4.3 Процесс монтажа и демонтажа ... 57
4.3.1 Монтажные размеры для устройств S7-1200 ... 57
4.3.2 Монтаж и демонтаж CPU ... 60
4.3.3 Монтаж и демонтаж SB, CB или BB ... 62
4.3.4 Монтаж и демонтаж SM ... 64
4.3.5 Монтаж и демонтаж CM или CP ... 66
4.3.6 Демонтаж и монтаж терминального блока S7-1200 ... 67
4.3.7 Монтаж и демонтаж расширительного кабеля ... 68
4.3.8 TS (TeleService) адаптер ... 70
4.3.8.1 Подключение TS адаптера ... 70
4.3.8.2 Установка SIM карты ... 71
4.3.8.3 Монтаж TS адаптера на DIN-рейку ... 72
4.3.8.4 Монтаж TS адаптера на пластину ... 73

4.4 Правила электромонтажа ... 74

5 Основы PLC ... 83

5.1 Выполнение программы пользователя ... 83
5.1.1 Рабочие состояния CPU ... 87
5.1.2 Обработка цикла в рабочем состоянии RUN ... 90
5.1.3 Организационные блоки (OB) ... 91
5.1.3.1 OB программного цикла ... 91
5.1.3.2 Пусковой OB ... 92
5.1.3.3 OB прерывания по задержке времени ... 92
5.1.3.4 OB циклических прерываний ... 93
5.1.3.5 OB аппаратных прерываний ... 94
5.1.3.6 OB прерывания по ошибке времени ... 95
5.1.3.7 OB диагностического прерывания ... 96
5.1.3.8 OB "Удаление или вставка модулей" ... 98
5.1.3.9 OB ошибки стойки или станции ("Rack or station failure OB") ... 99
5.1.3.10 OB прерывания по времени ... 99
5.1.3.11 OB состояния ... 100
5.1.3.12 OB прерывания по обновлению ... 101
5.1.3.13 OB прерывания по профилю ... 101
5.1.3.14 OB MC-Servo и MC-Interpolator ... 101
5.1.3.15 MC-PreServo ... 102
5.1.3.16 MC-PostServo ... 102
5.1.3.17 Приоритеты и лист ожидания для обработки событий ... 103
5.1.4 Контроль и конфигурирование времени цикла ... 107
5.1.5 Память CPU ... 109
5.1.5.1 Системная и тактовая память ... 111
5.1.6 Буфер диагностики ... 113
5.1.7 Часы реального времени ... 114
5.1.8 Конфигурирование выходов для перехода из RUN в STOP ... 115

5.2 Хранение данных, области памяти, I/O и адресация ... 116
5.2.1 Доступ к данным S7-1200 ... 116

5.3 Обработка аналоговых значений ... 122

5.4 Типы данных ... 123
5.4.1 Типы данных Bool, Byte, Word и DWord ... 125
5.4.2 Целочисленные типы данных ... 126
5.4.3 Типы данных для действительных чисел с плавающей запятой ... 126
5.4.4 Типы данных времени и даты ... 127
5.4.5 Символьные и строковые типы данных ... 129
5.4.6 Тип данных ARRAY ... 131
5.4.7 Структурированные типы данных ... 132
5.4.8 Тип данных PLC ... 132
5.4.9 Тип данных указателей "Variant" ... 133
5.4.10 Обращение к "срезу" тегированного типа данных ... 133
5.4.11 Обращение к переменной (тегу) с использование AT наложения ... 134

5.5 Использование карты памяти ... 136
5.5.1 Вставка карты памяти в CPU ... 137
5.5.2 Настройка параметров запуска CPU перед копированием проекта на карту памяти ... 139
5.5.3 Трансферная карта ... 140
5.5.4 Программная карта ... 142

5.5.5 Обновление прошивки ... 147
5.6 Порядок действий при утерянном пароле ... 149
6 Конфигурация устройства ... 151
6.1 Вставка CPU ... 152
6.2 Выгрузка конфигурации из подключенного CPU ... 154
6.3 Добавление модулей в конфигурацию ... 155
6.4 Управление конфигурацией ... 156
6.4.1 Преимущества и использование управления конфигурацией ... 156
6.4.2 Конфигурирование центральной системы и дополнительных модулей ... 157
6.4.3 Пример для управления конфигурацией ... 163
6.5 Изменение устройства ... 166
6.6 Конфигурирование работы CPU ... 167
6.6.1 Обзор ... 167
6.6.2 Установка времени фильтра для цифровых входов ... 169
6.6.3 Захват импульсов ... 170
6.7 Конфигурирование многоязыковой поддержки ... 172
6.8 Конфигурирование параметров модулей ... 174
6.9 Конфигурирование CPU для коммуникации ... 175
6.10 Синхронизация времени ... 176
7 Основы программирования ... 179
7.1 Руководства по разработке решения автоматизации с устройством PLC ... 179
7.2 Структурирование программы пользователя ... 180
7.3 Использование блоков для структурирования программы ... 182
7.3.1 Организационный блок (OB) ... 183
7.3.2 Функция (FC) ... 184
7.3.3 Функциональный блок (FB) ... 185
7.3.4 Блок данных (DB) ... 186
7.3.5 Создание повторно используемых блоков кода ... 188
7.3.6 Передача параметров в блоки ... 188
7.4 Целостность данных ... 191
7.5 Язык программирования ... 192
7.5.1 Релейно–контактная схема (LAD) ... 192
7.5.2 Функциональная блок-схема (FBD) ... 193
7.5.3 SCL ... 194
7.5.3.1 Редактор текстов программы SCL ... 194
7.5.3.2 Выражения и операции SCL ... 196
7.5.3.3 Индексированная адресация с помощью инструкций PEEK и POKE ... 199
7.5.4 EN и ENO в LAD, FBD и SCL ... 201
7.6 Защита ... 203
7.6.1 Защита от несанкционированного доступа для CPU ... 203
7.6.2 Внешняя загружаемая память ... 206
7.6.3 Защита ноу-хау ... 206
7.6.4 Защита от копирования ... 207

7.7 Загрузка элементов программы ... 209
7.8 Синхронизация Онлайн-CPU и Офлайн-проекта ... 212
7.9 Выгрузка из Онлайн-CPU ... 214
7.9.1 Сравнение Онлайн-CPU с Офлайн-CPU ... 214
7.10 Отладка и тестирование программы ... 215
7.10.1 Мониторинг и изменение данных в CPU ... 215
7.10.2 Таблицы текущего контроля и таблицы принудительно задаваемых значений ... 215
7.10.3 Перекрестная ссылка для отображения использования ... 216
7.10.4 Структура вызовов для проверки иерархии вызовов ... 217
8 Инструкции ... 219
8.1 Битовые логические операции ... 219
8.1.1 Инструкции битовой логики ... 219
8.1.2 Операции установки и сброса ... 222
8.1.3 Операции по положительному и отрицательному фронту ... 224
8.2 Принцип работы таймеров ... 227
8.3 Принцип работы счётчиков ... 236
8.4 Принцип работы компараторов ... 242
8.4.1 Операции сравнения ... 242
8.4.2 Инструкции IN_Range (значение в диапазоне) и OUT_Range (значение вне диапазона) ... 243
8.4.3 Инструкции OK (проверить на достоверность) и NOT_OK (проверить на недостоверность) ... 244
8.4.4 Операции сравнения типов данных Variant и Array ... 244
8.4.4.1 Операции сравнения по равенству и неравенству ... 244
8.4.4.2 Операции сравнения с нулевым значением ... 245
8.4.4.3 IS_ARRAY (проверка на МАССИВ) ... 246
8.5 Арифметические функции ... 247
8.5.1 CALCULATE (вычисление) ... 247
8.5.2 Инструкции сложения, вычитания, умножения и деления ... 248
8.5.3 MOD (остаток от деления) ... 249
8.5.4 NEG (создать двоичное дополнение) ... 250
8.5.5 INC (увеличение на 1) и DEC (уменьшение на 1) ... 251
8.5.6 ABS (создание абсолютного значения) ... 251
8.5.7 MIN (вычислить минимум) и MAX (вычислить максимум) ... 252
8.5.8 LIMIT (установка предельного значения) ... 253
8.5.9 Экспоненциальные, логарифмические и тригонометрические инструкции ... 254
8.6 Инструкции для передачи данных ... 256
8.6.1 MOVE (копирование значения), MOVE_BLK (копирование области), UMOVE_BLK (непрерывное копирование области) и MOVE_BLK_VARIANT (копирование области) ... 256
8.6.2 Десериализация ... 260
8.6.3 Serialize ... 263
8.6.4 Инструкции FILL_BLK (заполнить блок) и UFILL_BLK (заполнить блок без прерываний) ... 265
8.6.5 SWAP (поменять месторасположение) ... 267
8.6.6 LOWER_BOUND: (считывание нижней границы массива ARRAY) ... 268
8.6.7 UPPER_BOUND: (считывание верхней границы массива ARRAY) ... 269
8.6.8 Инструкции чтения / записи памяти ... 271

8.6.8.1 PEEK и POKE (только SCL) ... 271
8.6.8.2 Инструкции чтения и записи формата Big- и Little-Endian (SCL) ... 272
8.6.9 Инструкции Variant ... 274
8.6.9.1 VariantGet (чтение значения переменной (тега) VARIANT) ... 274
8.6.9.2 VariantPut (запись значения переменной (тега) VARIANT) ... 275
8.6.9.3 CountOfElements (запрос количества элементов массива ARRAY) ... 276
8.6.10 Унаследованные инструкции ... 277
8.6.10.1 Инструкции FieldRead (чтение поля) и FieldWrite (запись поля) ... 277
8.6.11 SCATTER ... 279
8.6.12 SCATTER_BLK ... 283
8.6.13 GATHER ... 288
8.6.14 GATHER_BLK ... 293

8.7 Операции преобразования ... 299
8.7.1 CONV (преобразование значения) ... 299
8.7.2 Инструкции преобразования в SCL ... 300
8.7.3 Инструкции ROUND (округление числа) и TRUNC (округление до целого числа) ... 303
8.7.4 CEIL и FLOOR (округление числа с плавающей запятой до следующего целого числа в сторону увеличения или уменьшения) ... 304
8.7.5 SCALE_X (масштабирование) и NORM_X (нормализация) ... 305
8.7.6 Инструкции преобразования типа Variant ... 308
8.7.6.1 VARIANT_TO_DB_ANY (преобразование из VARIANT в DB_ANY) ... 308
8.7.6.2 DB_ANY_TO_VARIANT (преобразование DB_Any в VARIANT) ... 309

8.8 Управление программой ... 311
8.8.1 Инструкции JMP (переход при RLO = 1), JMPN (переход при RLO = 0) и Label (метка перехода) ... 311
8.8.2 JMP_LIST (определение списка переходов) ... 312
8.8.3 SWITCH (распределение переходов) ... 313
8.8.4 RET (возвращаемое значение) ... 315
8.8.5 ENDIS_PW (включить/выключить пароль CPU) ... 316
8.8.6 RE_TRIGR (перезапуск времени контроля цикла) ... 318
8.8.7 STP (завершение программы) ... 319
8.8.8 Инструкции GET_ERROR и GET_ERROR_ID (локальное получение ошибки локальное получение ID ошибки) ... 320
8.8.9 RUNTIME (измерение времени выполнения программы) ... 323
8.8.10 Операторы управления программой в SCL ... 325
8.8.10.1 Обзор операторов управления программой в SCL ... 325
8.8.10.2 Оператор IF-THEN ... 326
8.8.10.3 Оператор CASE ... 327
8.8.10.4 Оператор FOR ... 329
8.8.10.5 Оператор WHILE-DO ... 330
8.8.10.6 Оператор REPEAT-UNTIL ... 331
8.8.10.7 Оператор CONTINUE ... 331
8.8.10.8 Оператор EXIT ... 332
8.8.10.9 Оператор GOTO ... 333
8.8.10.10 Оператор RETURN ... 333

8.9 Логические операции ... 334
8.9.1 Логические операции AND (И), OR (ИЛИ) и XOR (ИСКЛЮЧАЮЩЕЕ ИЛИ) ... 334
8.9.2 INV (создание дополнения до единицы) ... 335
8.9.3 Инструкции DECO (декодирование) и ENCO (кодирование) ... 335
8.9.4 Инструкции SEL (выбор), MUX (мультиплексирование) и DEMUX (демультиплексирование) ... 337

8.10 Сдвиг и циклический сдвиг ... 340
8.10.1 Инструкции SHR (сдвиг вправо) и SHL (сдвиг влево) ... 340
8.10.2 Инструкции ROR (циклический сдвиг вправо) и ROL (циклический сдвиг влево) ... 341

9 Расширенные инструкции ... 343

9.1 Функции даты, времени и часов ... 343
9.1.1 Инструкции для работы с датой и временем ... 343
9.1.2 Функции часов ... 346
9.1.3 Структура данных TimeTransformationRule ... 348
9.1.4 Инструкция SET_TIMEZONE (установка часового пояса) ... 349
9.1.5 RTM (счетчик часов работы) ... 350

9.2 Строчные и символьные инструкции ... 352
9.2.1 Тип данных String (строка) ... 352
9.2.2 S_MOV (перемещение строки символов) ... 353
9.2.3 Инструкции для преобразования строк символов ... 353
9.2.3.1 Инструкции S_CONV, STRG_VAL и VAL_STRG (преобразование из/в строку символов и числовое значение) ... 353
9.2.3.2 Инструкции Strg_TO_Chars и Chars_TO_Strg (преобразование в(из) строку символов и массив из CHAR) ... 362
9.2.3.3 ATH и HTA (преобразование в/из строку символов ASCII и шестнадцатеричное число) ... 364
9.2.4 Строковые инструкции ... 366
9.2.4.1 MAX_LEN (максимальная длина строки символов) ... 367
9.2.4.2 LEN (определение длины строки символов) ... 367
9.2.4.3 CONCAT (объединение строк символов) ... 368
9.2.4.4 Инструкции LEFT, RIGHT и MID (чтение подстрок в строке символов) ... 369
9.2.4.5 Инструкция DELETE (удаление символов из строки символов) ... 370
9.2.4.6 INSERT (вставка символов в строку символов) ... 371
9.2.4.7 REPLACE (замена символов в строке символов) ... 372
9.2.4.8 FIND (поиск символов в строке символов) ... 373
9.2.5 Информация при выполнении ... 374
9.2.5.1 GetSymbolName (считывание имени переменной во входном параметре) ... 374
9.2.5.2 GetSymbolPath (запрос составного глобального имени входного параметра) ... 377
9.2.5.3 GetInstanceName (считывание имени экземпляра блока) ... 380
9.2.5.4 GetSymbolPath (запрос составного глобального имени экземпляра блока) ... 382
9.2.5.5 GetBlockName (считывание имени блока) ... 384

9.3 Распределенный ввод/вывод (PROFINET, PROFIBUS или AS-i) ... 386
9.3.1 Инструкции для распределенного ввода/вывода ... 386
9.3.2 Инструкции RDREC и WRREC (чтение/запись набора данных) ... 387
9.3.3 GETIO (считывание образа процесса) ... 390
9.3.4 SETIO (передача образа процесса) ... 391
9.3.5 GETIO_PART (считывание области образа процесса) ... 392
9.3.6 SETIO_PART (передача области образа процесса) ... 394
9.3.7 Инструкция RALRM (получение прерывания) ... 395
9.3.8 D_ACT_DP (деактивация/активация ведомых устройств DP) ... 399
9.3.9 Параметр STATUS для RDREC, WRREC и RALRM ... 404
9.3.10 Другие ... 409
9.3.10.1 DPRD_DAT und DPWR_DAT (чтение/запись непротиворечивых данных) ... 409
9.3.10.2 RCVREC (получение набора данных) ... 412
9.3.10.3 PRVREC (предоставление набора данных) ... 415
9.3.10.4 DPNRM_DG (чтение диагностических данных ведомого устройства DP) ... 417

9.4 PROFIenergy ... 420
9.5 Прерывания ... 421
9.5.1 Инструкции ATTACH и DETACH (согласование OB и прерывающего события/отмена согласования) ... 421
9.5.2 Циклические прерывания ... 424
9.5.2.1 SET_CINT (параметрирование циклического прерывания) ... 424
9.5.2.2 QRY_CINT (запрос параметров циклического прерывания) ... 426
9.5.3 Прерывания по времени ... 428
9.5.3.1 SET_TINTL (установка прерывания по времени) ... 428
9.5.3.2 CAN_TINT (удаление прерывания по времени) ... 430
9.5.3.3 ACT_TINT (активация прерывания по времени) ... 430
9.5.3.4 QRY_TINT (запрос состояния прерывания по времени) ... 431
9.5.4 Прерывания по задержке времени ... 432
9.5.5 Инструкции DIS_AIRT und EN_AIRT (задержка/активация выполнения прерываний с более высоким приоритетом и асинхронных ошибок) ... 435
9.6 Прерывания ... 436
9.6.1 Gen_UsrMsg (создание диагностических сообщений пользователя) ... 436
9.7 Диагностика (PROFINET или PROFIBUS) ... 439
9.7.1 Диагностические инструкции ... 439
9.7.2 RD_SINFO (считывание начальной информации текущего блока OB) ... 440
9.7.3 LED (чтение состояния светодиода) ... 449
9.7.4 Get_IM_Data (чтение данных идентификации и обслуживания) ... 450
9.7.5 Get_Name (чтение имени устройства PROFINET IO) ... 452
9.7.6 GetStationInfo (чтение IP- или MAC-адреса устройства PROFINET IO) ... 458
9.7.7 Инструкция DeviceStates ... 468
9.7.7.1 Примеры для конфигурации DeviceStates ... 469
9.7.8 Инструкция ModuleStates ... 474
9.7.8.1 Примеры для конфигурации ModuleStates ... 475
9.7.9 GET_DIAG (чтение информации системы диагностики) ... 479
9.7.10 Диагностические события для распределенной периферии ... 486
9.8 Импульс ... 487
9.8.1 CTRL_PWM (широтно-импульсная модуляция) ... 487
9.8.2 CTRL_PTO (вывод последовательности импульсов) ... 488
9.8.3 Принцип работы для импульсных выходов ... 492
9.8.4 Конфигурирование импульсного канала для PWM или PTO ... 494
9.9 Рецептуры и архивы данных ... 499
9.9.1 Рецептуры ... 499
9.9.1.1 Обзор рецептур ... 499
9.9.1.2 Пример для рецептуры ... 500
9.9.1.3 Программные инструкция для передачи рецептурных данных ... 504
9.9.1.4 Пример программы для работы с рецептурами ... 508
9.9.2 Архивы данных ... 510
9.9.2.1 Структура набора данных архива ... 511
9.9.2.2 Программные инструкции для управления архивами данных ... 512
9.9.2.3 Работа с архивами данных ... 527
9.9.2.4 Ограничение размера для файлов архивов данных ... 528
9.9.2.5 Пример программы для архивов данных ... 532
9.10 Управление блоком данных ... 537
9.10.1 CREATE_DB (создание блока данных) ... 537

9.10.2 Инструкции READ_DBL и WRIT_DBL (чтение/запись блока данных в загружаемой памяти) ... 541
9.10.3 ATTR_DB (чтение атрибутов блока данных) ... 544
9.10.4 DELETE_DB (удаление блока данных) ... 546

9.11 Обработка адресов ... 548
9.11.1 GEO2LOG (определение аппаратного идентификатора из слота) ... 548
9.11.2 LOG2GEO (определение слота по аппаратному идентификатору) ... 549
9.11.3 IO2MOD (определение аппаратного идентификатора по адресу ввода/вывода) ... 551
9.11.4 RD_ADDR (определение адреса ввода/вывода по аппаратному идентификатору) ... 552
9.11.5 Тип системных данных GEOADDR ... 553

9.12 Общие коды ошибок для расширенных инструкций ... 554

10 Технологические инструкции ... 557

10.1 Счет (высокоскоростные счетчики) ... 557
10.1.1 Инструкции CTRL_HSC_EXT (управление высокоскоростным счетчиком) ... 558
10.1.1.1 Обзор инструкции ... 558
10.1.1.2 Пример ... 559
10.1.1.3 Типы системных данных (SDT) инструкции CTRL_HSC_EXT ... 562
10.1.2 Работа высокоскоростного счётчика ... 567
10.1.2.1 Функция синхронизации ... 567
10.1.2.2 Функция ограничения счета ... 568
10.1.2.3 Функция захвата ... 570
10.1.2.4 Функция сравнения ... 572
10.1.2.5 Применение ... 572
10.1.3 Конфигурирование высокоскоростного счетчика ... 573
10.1.3.1 Типы счета ... 575
10.1.3.2 Рабочая фаза ... 576
10.1.3.3 Начальные значения ... 580
10.1.3.4 Функции входов ... 581
10.1.3.5 Функция выхода ... 582
10.1.3.6 Прерывающие события ... 583
10.1.3.7 Разводка соединителей аппаратного входа ... 583
10.1.3.8 Разводка соединителей аппаратного выхода ... 585
10.1.3.9 Адреса ввода HSC ... 586
10.1.3.10 Аппаратный идентификатор ... 586
10.1.4 Старая инструкция CTRL_HSC (управление высокоскоростным счетчиком) ... 587
10.1.4.1 Обзор инструкции ... 587
10.1.4.2 Использование CTRL_HSC ... 589
10.1.4.3 Текущее состояние счетчика HSC ... 589

10.2 ПИД-регулирование ... 590
10.2.1 Вставка ПИД инструкции и технологического объекта ... 592
10.2.2 PID_Compact ... 594
10.2.2.1 Инструкция PID_Compact ... 594
10.2.2.2 Предельные значения для технологического значения инструкции PID_Compact ... 597
10.2.2.3 Параметр ErrorBit инструкции PID_Compact ... 599
10.2.2.4 Параметры предупреждения инструкции PID_Compact ... 601
10.2.3 PID_3Step ... 602
10.2.3.1 Инструкция PID_3Step ... 602
10.2.3.2 Параметр ErrorBit инструкции PID_3Step ... 610
10.2.3.3 Параметры предупреждения инструкции PID_3Step ... 612
10.2.4 PID_Temp ... 613

10.2.4.1 Инструкция PID_Temp ... 613
10.2.4.2 Параметр ErrorBit из PID_Temp ... 624
10.2.4.3 Параметры предупреждений PID_Temp ... 626
10.2.5 Конфигурирование регуляторов PID_Compact и PID_3Step ... 626
10.2.6 Конфигурирование регулятора PID_Temp ... 629
10.2.7 Ввод в эксплуатацию регуляторов PID_Compact и PID_3Step ... 644
10.2.8 Ввод в эксплуатацию регулятора PID_Temp ... 646

10.3 Управление перемещением: ... 656
10.3.1 Фазирование ... 662
10.3.2 Конфигурирование генератора импульсов ... 665
10.3.3 Нерегулируемое управление перемещением ... 666
10.3.3.1 Конфигурирование оси ... 666
10.3.3.2 Ввод в эксплуатацию ... 670
10.3.4 Регулируемое управление перемещением ... 675
10.3.4.1 Конфигурирование оси ... 675
10.3.4.2 ServoOB ... 682
10.3.4.3 Режим управления по скорости ... 684
10.3.4.4 Поддержка телеграммы 4 ... 686
10.3.4.5 Моделируемая ось ... 691
10.3.4.6 Перенос данных ... 693
10.3.4.7 Управление осью с помощью технологического модуля (TM) Pulse ... 704
10.3.5 Конфигурирование TO_CommandTable_PTO ... 710
10.3.6 Принцип работы управления перемещением для S7-1200 ... 714
10.3.6.1 Используемые для управления перемещением выходы CPU ... 714
10.3.6.2 Аппаратные и программные конечные выключатели для управления перемещением ... 716
10.3.6.3 Движение к референтной точке (реферирование) ... 726
10.3.6.4 Ограничение рывка ... 735
10.3.7 Инструкции управления перемещением ... 735
10.3.7.1 Обзор MC-инструкций ... 735
10.3.7.2 Инструкция MC_Power (деблокировка/блокировка оси) ... 737
10.3.7.3 Инструкция MC_Reset (подтвердить ошибку) ... 740
10.3.7.4 Инструкция MC_Home (выполнение движения оси к референтной точке) ... 741
10.3.7.5 Инструкция MC_Halt (приостановка оси) ... 744
10.3.7.6 MC_MoveAbsolute (абсолютное позиционирование оси) ... 746
10.3.7.7 Инструкция MC_MoveRelative (относительное позиционирование оси) ... 749
10.3.7.8 Инструкция MC_MoveVelocity (перемещение оси с заданной скоростью) ... 751
10.3.7.9 Инструкция MC_MoveJog (перемещение оси в толчковом режиме) ... 754
10.3.7.10 Инструкция MC_CommandTable (выполнение команд управления осью как последовательности движений) ... 756
10.3.7.11 Инструкция MC_ChangeDynamic (изменение настроек динамики для оси) ... 759
10.3.7.12 Инструкция MC_WriteParam (запись параметров технологического объекта) ... 761
10.3.7.13 Инструкция MC_ReadParam (чтение параметров технологического объекта) ... 763
10.3.8 Контроль активных команд ... 765
10.3.8.1 Контроль инструкции управления перемещением с помощью выходного параметра "Done" ... 765
10.3.8.2 Мониторинг MC_Velocity ... 770
10.3.8.3 Мониторинг инструкции MC_MoveJog ... 774
10.3.9 Идентификаторы ошибок (ErrorID) и информационные сообщения об ошибках (ErrorInfo) для управления перемещением ... 777

11 Коммуникация ... 805

11.1 Асинхронные коммуникационные соединения ... 807

11.2 PROFINET ... 810
11.2.1 Создание сетевого соединения ... 811
11.2.2 Конфигурирование пути соединения между локальным и партнерским CPU ... 812
11.2.3 Назначение IP-адресов ... 815
11.2.3.1 Назначение IP-адресов программатору и сетевым устройствам ... 815
11.2.3.2 Проверка IP-адреса программатора ... 817
11.2.3.3 Назначение IP-адресов для CPU в режиме онлайн ... 817
11.2.3.4 Конфигурирование IP-адреса для CPU в проекте ... 819
11.2.4 Тестирование сети PROFINET ... 823
11.2.5 Определение Ethernet-адреса (MAC-Adresse) у CPU ... 824
11.2.6 Конфигурирование NTP синхронизации (Network Time Protocol, NTP) ... 826
11.2.7 Пусковой период, присвоение имени и назначение адреса PROFINET устройствам ... 827
11.2.8 Открытые коммуникационные соединения пользователя ... 828
11.2.8.1 Протоколы ... 828
11.2.8.2 TCP и ISO on TCP ... 829
11.2.8.3 Коммуникационные службы и используемые номера портов ... 830
11.2.8.4 Ad hoc режим ... 831
11.2.8.5 Идентификаторы (ID) для открытых коммуникационных соединений пользователя ... 832
11.2.8.6 Параметры для соединения PROFINET ... 835
11.2.8.7 Конфигурирование DNS ... 841
11.2.8.8 Инструкции TSEND_C и TRCV_C ... 842
11.2.8.9 Инструкции TSEND_C и TRCV_C в старых системах ... 856
11.2.8.10 Инструкции TCON, TDISCON, TSEND и TRCV ... 863
11.2.8.11 Старые инструкции TCON, TDISCON, TSEND и TRCV ... 875
11.2.8.12 Инструкция T_RESET (разорвать и восстановить существующее соединение) ... 885
11.2.8.13 Инструкция T_DIAG (проверить состояние соединения и считать информацию) ... 887
11.2.8.14 Инструкция TMAIL_C (передача Email через Ethernet интерфейс CPU) ... 891
11.2.8.15 UDP ... 913
11.2.8.16 TUSEND и TURCV ... 913
11.2.8.17 T_CONFIG ... 919
11.2.8.18 Общие параметры для инструкций ... 930
11.2.9 Обмен данными с программатором ... 931
11.2.9.1 Установление аппаратного коммуникационного соединения ... 932
11.2.9.2 Конфигурирование устройств ... 932
11.2.9.3 Назначение IP-адресов ... 933
11.2.9.4 Тестирование сети PROFINET ... 933
11.2.10 Коммуникация HMI/PLC ... 934
11.2.10.1 Конфигурирование логических сетевых соединений между двумя устройствами ... 935
11.2.11 Коммуникация PLC/PLC ... 935
11.2.11.1 Конфигурирование логических сетевых соединений между двумя устройствами ... 936
11.2.11.2 Конфигурирование пути соединения между локальным и партнерским CPU ... 937
11.2.11.3 Конфигурирование параметров передачи и приема ... 937
11.2.12 Конфигурирование CPU и устройства PROFINET IO ... 940
11.2.12.1 Добавление устройства PROFINET IO ... 940
11.2.12.2 Присвоение имен CPU и устройствам ... 941
11.2.12.3 Назначение IP-адресов ... 941
11.2.12.4 Конфигурирование времени цикла IO ... 942
11.2.13 Конфигурирование CPU и интеллектуального устройства PROFINET ... 943
11.2.13.1 Функциональные возможности I устройства ... 943

11.2.13.2 Свойства и преимущества I-устройства ... 944
11.2.13.3 Отличительные особенности I-устройства ... 944
11.2.13.4 Обмен данными между системами ввода-вывода верхнего и нижнего уровня ... 947
11.2.13.5 Конфигурирование I-устройства ... 950
11.2.14 Устройства общего доступа ... 952
11.2.14.1 Функциональные возможности устройства общего доступа ... 952
11.2.14.2 Пример: Конфигурирование устройства общего доступа (GSD конфигурация) ... 956
11.2.14.3 Пример: Конфигурирование I-устройства как устройства общего доступа ... 961
11.2.15 Протокол резервирования среды (MRP) ... 971
11.2.15.1 Резервирование среды в кольцевых топологиях ... 971
11.2.15.2 Использование протокола резервирования среды (MRP) ... 972
11.2.15.3 Конфигурирование резервирования среды ... 975
11.2.16 S7-маршрутизация ... 978
11.2.16.1 S7-маршрутизация между CPU и CP-интерфейсами ... 979
11.2.16.2 S7-маршрутизация между двумя интерфейсами CP ... 980
11.2.17 Отключение SNMP ... 980
11.2.17.1 Отключение SNMP ... 981
11.2.18 Диагностика ... 983
11.2.19 Инструкции для распределенной периферии ... 983
11.2.20 Диагностические инструкции ... 983
11.2.21 Диагностические события для распределенной периферии ... 983

11.3 PROFIBUS ... 984
11.3.1 Коммуникационные службы PROFIBUS-CM ... 985
11.3.2 Ссылка на руководства пользователя для PROFIBUS-CM ... 986
11.3.3 Конфигурирование ведущего и ведомого устройства DP ... 987
11.3.3.1 Добавление CM 1243-5 (ведущее устройство DP) и ведомого устройства DP ... 987
11.3.3.2 Конфигурирование логических сетевых соединений между двумя устройствами PROFIBUS ... 988
11.3.3.3 Назначение адресов PROFIBUS для CM 1243-5 и ведомого устройства DP ... 988
11.3.4 Инструкции для распределенной периферии ... 990
11.3.5 Диагностические инструкции ... 990
11.3.6 Диагностические события для распределенной периферии ... 990

11.4 ASi ... 990
11.4.1 Конфигурирование ведущего и ведомого AS-i устройств ... 991
11.4.1.1 Добавление модуля ведущего AS-i устройства CM 1243-2 и ведомого AS-i устройства ... 992
11.4.1.2 Конфигурирование логических сетевых соединений между двумя устройствами AS-i. ... 992
11.4.1.3 Конфигурирование свойств ведущего AS-i устройства CM1243-2 ... 993
11.4.1.4 Назначение AS-i адреса ведомому AS-i устройству ... 994
11.4.2 Обмен данными между программой пользователя и ведомыми AS-i устройствами ... 996
11.4.2.1 Базовая конфигурация в STEP 7 ... 996
11.4.2.2 Конфигурирование ведомых устройств с помощью STEP 7 ... 998
11.4.3 Инструкции для распределенной периферии ... 1000
11.4.4 Работа с онлайн-инструментами AS-i ... 1000

11.5 Коммуникация S7 ... 1002
11.5.1 Инструкции GET и PUT (чтение и запись из удаленного CPU) ... 1002
11.5.2 Создание соединения S7 ... 1006
11.5.3 Конфигурирование пути соединения между локальным и партнерским CPU ... 1007
11.5.4 Назначение параметров соединения для GET/PUT ... 1007
11.5.4.1 Параметры соединения ... 1008

11.5.4.2 Конфигурирование S7-соединения между CPU ... 1011
11.6 Что делать, если доступ к CPU по IP-адресу невозможен ... 1016
11.7 Сервер OPC UA ... 1017
11.7.1 Конфигурирование сервера OPC UA ... 1017
11.7.1.1 Активация сервера OPC UA ... 1017
11.7.1.2 Настройки для сервера OPC UA ... 1018
11.7.2 Безопасность сервера OPC UA ... 1018
11.7.2.1 Поддерживаемые правила безопасности ... 1018
11.7.2.2 Доверенные клиенты ... 1019
11.7.2.3 Аутентификация пользователя ... 1019
11.7.3 Интерфейс сервера OPC UA ... 1019
11.7.3.1 Поддерживаемые типы данных ... 1020
11.7.3.2 Представление PLC ... 1020
11.7.3.3 Загружаемые интерфейсы сервера ... 1021
12 Веб-сервер ... 1023
12.1 Активация веб-сервера ... 1026
12.2 Конфигурирование пользователей веб-сервера ... 1028
12.3 Доступ к веб-страницам от PC ... 1029
12.4 Доступ к веб-страницам через мобильное устройство ... 1031
12.5 Использование CP модуля для доступа к веб-страницам ... 1033
12.6 Загрузка и установка сертификата безопасности ... 1033
12.7 Стандартные веб-страницы ... 1036
12.7.1 Компоновка стандартных веб-страниц ... 1036
12.7.2 Базовые страницы ... 1037
12.7.3 Вход в систему и права пользователя ... 1038
12.7.4 Вводная страница ... 1041
12.7.5 Начальная страница ... 1042
12.7.6 Диагностика ... 1043
12.7.7 Буфер диагностики ... 1045
12.7.8 Информация о модулях ... 1046
12.7.9 Коммуникация ... 1050
12.7.10 Состояние переменных (тегов) ... 1055
12.7.11 Таблицы текущего контроля ... 1056
12.7.12 Резервное онлайн-копирование ... 1059
12.7.13 Страница архивов данных ... 1061
12.7.14 Файлы пользователя ... 1064
12.7.15 API для архивов данных и файлов пользователя ... 1068
12.7.16 Браузер файлов ... 1068
12.8 Пользовательские веб-страницы ... 1070
12.8.1 Создание HTML-страниц ... 1071
12.8.2 Поддерживаемые веб-сервером S7-1200 команды AWP ... 1072
12.8.2.1 Чтение переменных ... 1074
12.8.2.2 Запись переменных ... 1075
12.8.2.3 Чтение специальных переменных ... 1077
12.8.2.4 Запись специальных переменных ... 1079
12.8.2.5 Использование псевдонимов для ссылки на переменные ... 1080
12.8.2.6 Определение типов перечисления ... 1081

12.8.2.7 Ссылка на переменные CPU с типом перечислений ... 1082
12.8.2.8 Создание фрагментов ... 1083
12.8.2.9 Импорт фрагментов ... 1084
12.8.2.10 Объединение определений ... 1085
12.8.2.11 Обработка имен переменных (тегов), содержащих специальные символы ... 1085
12.8.3 Конфигурирование использования пользовательских веб-страниц ... 1087
12.8.4 Конфигурирование страницы входа ... 1089
12.8.5 Программирование WWW инструкции для пользовательских веб-страниц ... 1089
12.8.6 Загрузка программных блоков в CPU ... 1091
12.8.7 Доступ к пользовательским веб-страницам ... 1092
12.8.8 Ограничения для пользовательских веб-страниц ... 1092
12.8.9 Пример для пользовательской веб-страницы ... 1094
12.8.9.1 Веб-страница для мониторинга и управления ветряной турбиной ... 1094
12.8.9.2 Чтение и отображение данных контроллера ... 1096
12.8.9.3 Использование типа перечисления ... 1097
12.8.9.4 Запись введенных пользователем данных в контроллер ... 1098
12.8.9.5 Запись специальных переменных ... 1099
12.8.9.6 Справочная информация: HTML-код веб-страницы "Remote Wind Turbine Monitor" ... 1099
12.8.9.7 Пример конфигурирования веб-страницы в STEP 7 ... 1103
12.8.10 Создание пользовательских веб-страниц на нескольких языках ... 1105
12.8.10.1 Создание структуры папок ... 1105
12.8.10.2 Программирование переключения языка ... 1105
12.8.10.3 Конфигурирование STEP 7 для использования многоязыковой структуры страницы. 1108
12.8.11 Расширенное управление пользовательскими веб-страницами ... 1108

12.9 Ограничения ... 1112
12.9.1 Использование JavaScript ... 1113
12.9.2 Ограничения функциональности, когда интернет-опции отключают куки ... 1113
12.9.3 Правила ввода имен переменных (тегов) и значений ... 1114
12.9.4 Импорт архивов данных в формате CSV в не американские/английские версии Microsoft Excel ... 1114

13 Коммуникационный процессор и Modbus TCP ... 1117

13.1 Использование последовательных коммуникационных интерфейсов ... 1117

13.2 Включение оконечной нагрузки в шинном разъеме RS485 ... 1118

13.3 Коммуникации точка-точка (PtP) ... 1119
13.3.1 PtP, свободно программируемая коммуникация ... 1120
13.3.2 3964(R)-коммуникация ... 1122
13.3.3 Конфигурирование свободно программируемой PtP коммуникации ... 1123
13.3.3.1 Организация управления потоком ... 1125
13.3.3.2 Конфигурирование параметров передачи ... 1127
13.3.3.3 Конфигурирование параметров приема ... 1128
13.3.4 Конфигурирование 3964(R) коммуникации ... 1136
13.3.4.1 Конфигурирование 3964(R) коммуникационных портов ... 1136
13.3.4.2 Конфигурирование приоритета 3964(R) и параметров протокола ... 1137
13.3.5 Инструкции точка-точка ... 1139
13.3.5.1 Общие параметры для операций точка-точка ... 1139
13.3.5.2 Инструкция Port_Config (динамическое конфигурирование коммуникационных параметров) ... 1141
13.3.5.3 Инструкция Send_Config (динамическое конфигурирование параметров для последовательной коммуникации) ... 1144

13.3.5.4 Инструкция Receive_Config (динамическое конфигурирование параметров для последовательного приема) ... 1147
13.3.5.5 P3964_Config (конфигурирование протокола 3964(R)) ... 1152
13.3.5.6 Инструкция Send_P2P (отправить данные из передающего буфера) ... 1154
13.3.5.7 Инструкция Receive_P2P (разрешить прием сообщений) ... 1158
13.3.5.8 Инструкция Receive_Reset (удалить данные в принимающем буфере) ... 1160
13.3.5.9 Инструкция Signal_Get (запрос сигналов RS-232) ... 1161
13.3.5.10 Инструкция Signal_Set (установка сигналов RS-232) ... 1162
13.3.5.11 Get_Features ... 1163
13.3.5.12 Set_Features ... 1164
13.3.6 Программирование PtP-коммуникации ... 1165
13.3.6.1 Архитектура опроса ... 1166
13.3.7 Пример: Коммуникации точка-точка ... 1167
13.3.7.1 Конфигурирование коммуникационного модуля ... 1168
13.3.7.2 Рабочие режимы RS422 и RS485 ... 1171
13.3.7.3 Создание программы STEP 7 ... 1174
13.3.7.4 Конфигурирование эмулятора терминала ... 1175
13.3.7.5 Выполнение программы-примера ... 1176

13.4 Коммуникация через универсальный последовательный интерфейс (USS) ... 1176
13.4.1 Выбор версии USS-инструкций ... 1179
13.4.2 Необходимые условия для использования протокола USS ... 1180
13.4.3 Инструкции USS ... 1182
13.4.3.1 USS_Port_Scan (обработка коммуникации через сеть USS) ... 1182
13.4.3.2 Инструкция USS_Drive_Control (обмен данными с приводом) ... 1184
13.4.3.3 Инструкция USS_Read_Param (считывание параметров из привода) ... 1187
13.4.3.4 Инструкция USS_Write_Param (изменить параметры в приводе) ... 1188
13.4.4 Коды состояния USS ... 1190
13.4.5 Общие требования USS по настройке привода ... 1192
13.4.6 Пример: Общее USS-подключение и настройка привода ... 1192

13.5 Коммуникация Modbus ... 1195
13.5.1 Обзор коммуникация с помощью MODBUS RTU и MODBUS TCP ... 1195
13.5.2 Modbus TCP ... 1198
13.5.2.1 Обзор ... 1198
13.5.2.2 Выбор версии Modbus TCP инструкций ... 1199
13.5.2.3 Инструкции Modbus TCP ... 1200
13.5.2.4 Примеры для Modbus TCP ... 1262
13.5.3 Modbus RTU ... 1267
13.5.3.1 Обзор ... 1267
13.5.3.2 Выбор версии инструкций Modbus RTU ... 1269
13.5.3.3 Максимальное число поддерживаемых ведомых устройств Modbus ... 1269
13.5.3.4 Инструкции Modbus RTU ... 1270
13.5.3.5 Примеры Modbus RTU ... 1293

13.6 PtP коммуникация в старых системах (только CM/CB 1241) ... 1297
13.6.1 Старые инструкции точка-точка ... 1298
13.6.1.1 PORT_CFG (динамическое конфигурирование коммуникационных параметров) ... 1298
13.6.1.2 SEND_CFG (конфигурация передачи) ... 1300
13.6.1.3 RCV_CFG (конфигурация приема) ... 1301
13.6.1.4 SEND_PTP (отправить данные из передающего буфера) ... 1307
13.6.1.5 RCV_PTP (разрешить прием сообщений) ... 1309
13.6.1.6 RCV_RST (удалить данные в принимающем буфере) ... 1311
13.6.1.7 SGN_GET (запросить сигналы RS232) ... 1312

13.6.1.8 SGN_SET (установить сигналы RS-232) ... 1313

13.7 USS коммуникация в старых системах (только CM/CB 1241) ... 1314
13.7.1 Выбор версии USS-инструкций ... 1315
13.7.2 Необходимые условия для использования протокола USS ... 1316
13.7.3 Старые инструкции USS ... 1318
13.7.3.1 USS_PORT (обработка коммуникации через сеть USS) ... 1318
13.7.3.2 Инструкция USS_DRV (обмен данными с приводом) ... 1320
13.7.3.3 Инструкция USS_RPM (считывание параметров из привода) ... 1323
13.7.3.4 Инструкция USS_WPM (изменение параметров в приводе) ... 1324
13.7.4 Старые коды состояния USS ... 1326
13.7.5 Общие требования при настройке привода с использованием старой инструкции USS ... 1328

13.8 Modbus TCP коммуникация в старых системах ... 1328
13.8.1 Обзор ... 1328
13.8.2 Выбор версии Modbus TCP инструкций ... 1328
13.8.3 Старые инструкции Modbus TCP ... 1329
13.8.3.1 MB_CLIENT (обмен данными с использованием PROFINET в качестве клиента Modbus TCP) ... 1329
13.8.3.2 MB_SERVER (обмен данными с использованием PROFINET в качестве сервера Modbus TCP) ... 1335
13.8.4 Старые примеры Modbus TCP ... 1341
13.8.4.1 Пример: MB_SERVER для нескольких соединений TCP в старых системах ... 1341
13.8.4.2 Пример: MB_CLIENT 1 в старых системах: Несколько запросов с общим TCP соединением ... 1342
13.8.4.3 Пример: MB_CLIENT 2 в старых системах: Несколько запросов с различными TCP соединениями ... 1343
13.8.4.4 Пример: MB_CLIENT 3 в старых системах: Запрос записи для образа процесса выходов ... 1344
13.8.4.5 Пример: MB_CLIENT 4 в старых системах: Координация нескольких запросов ... 1345

13.9 Modbus RTU коммуникация в старых системах (только CM/CB 1241) ... 1346
13.9.1 Обзор ... 1346
13.9.2 Выбор версии инструкций Modbus RTU ... 1346
13.9.3 Старые инструкции Modbus RTU ... 1347
13.9.3.1 MB_COMM_LOAD (конфигурирование порта на PtP модуле для Modbus RTU) ... 1347
13.9.3.2 MB_MASTER (обмен данными через PtP-порт в качестве ведущего устройства Modbus RTU) ... 1350
13.9.3.3 MB_SLAVE (обмен данными через PtP-порт в качестве ведомого устройства Modbus RTU) ... 1356
13.9.4 Пример для Modbus RTU в старых системах ... 1363
13.9.4.1 Пример: Пример программы для ведущего устройства Modbus RTU в старых системах ... 1363
13.9.4.2 Пример: Пример программы для ведомого устройства Modbus RTU в старых системах ... 1365

13.10 Удаленная передача данных в промышленности (Industrial Remote Communication, IRC) ... 1366
13.10.1 Обзор коммуникационных процессоров для телеуправления ... 1366
13.10.2 Подключение к сети GSM ... 1369
13.10.3 Приложения для CP 1242-7 ... 1370
13.10.4 Другие возможности CP 1242-7 ... 1372
13.10.5 Дополнительная информация ... 1372

13.10.6 Аксессуары ... 1372
13.10.7 Примеры конфигурации для телеуправления ... 1373

14 Коммуникации TeleService (SMTP email) ... 1379

14.1 Инструкция TM_Mail (передача электронного сообщения) ... 1379

15 Онлайн инструменты и диагностика ... 1387

15.1 Светодиодные индикаторы состояния ... 1387
15.2 Установление онлайн-соединения с CPU ... 1390
15.3 Присвоение имени устройству PROFINET IO в режиме онлайн ... 1391
15.4 Установка IP адреса и времени суток ... 1393
15.5 Сброс на заводские настройки ... 1393
15.6 Обновление прошивки ... 1394
15.7 Форматирование карты памяти SIMATIC через STEP 7 ... 1396
15.8 Панель оператора для Онлайн-CPU ... 1396
15.9 Мониторинг времени цикла и использования ресурсов памяти ... 1397
15.10 Отображение диагностических событий в CPU ... 1397
15.11 Сравнение офлайн и онлайн CPU ... 1399
15.12 Сравнение топологии онлайн/офлайн ... 1400
15.13 Мониторинг и изменение значений в CPU ... 1401
15.13.1 Переход в онлайн для наблюдения за значениями в CPU ... 1401
15.13.2 Отображение состояния в редакторе текстов программ ... 1402
15.13.3 Регистрация мгновенных онлайн-значений DB для восстановления значений ... 1403
15.13.4 Мониторинг и изменение значений в CPU через таблицу текущего контроля ... 1404
15.13.4.1 Использование триггера при мониторинге или изменении переменных (тегов) PLC. 1406
15.13.4.2 Разрешение выходов в режиме STOP ... 1407
15.13.5 Принудительная установка значений в CPU ... 1407
15.13.5.1 Работа с таблицей принудительно задаваемых значений ... 1407
15.13.5.2 Принцип работы функции принудительной установки ... 1409
15.14 Загрузка в рабочем состоянии RUN ... 1410
15.14.1 Условия для "Загрузки в рабочем состоянии RUN" ... 1412
15.14.2 Редактирование программы в рабочем состоянии RUN ... 1413
15.14.3 Загрузка выбранных блоков ... 1414
15.14.4 Загрузка одиночного выбранного блока с ошибкой компиляции в другом блоке ... 1415
15.14.5 Изменение и загрузка существующих блоков в целевую систему в рабочем состоянии RUN ... 1416
15.14.6 Реакция системы на неудачный процесс загрузки ... 1419
15.14.7 Критерии безопасности при загрузке в рабочем состоянии RUN ... 1419
15.15 Трассировка и запись данных CPU по условиям запуска ... 1421
15.16 Определение типа обрыва провода с помощью модуля SM 1231 ... 1423
15.17 Резервное копирование и восстановление CPU ... 1426
15.17.1 Опции для резервного копирования и восстановления ... 1426
15.17.2 Резервное копирование Онлайн-CPU ... 1428
15.17.3 Восстановление CPU ... 1430

A Технические данные ... 1433

A.1 Веб-сайт Siemens для онлайн-поддержки ... 1433

A.2 Общие технические данные ... 1433

A.3 Разводка контактов PROFINET интерфейсный порт X1 ... 1443

A.4 CPU 1211C ... 1444
A.4.1 Общие технические данные и характеристики ... 1444
A.4.2 Поддерживаемые CPU 1211C таймеры, счётчики и блоки кода ... 1446
A.4.3 Цифровые входы и выходы ... 1450
A.4.4 Аналоговые входы ... 1452
A.4.4.1 Реакция на скачок встроенных аналоговых входов CPU ... 1453
A.4.4.2 Время выборки для встроенных аналоговых портов CPU ... 1453
A.4.4.3 Диапазоны измерений для аналоговых входов напряжения (CPU) ... 1453
A.4.5 Схемы электрических соединений CPU 1211C ... 1454

A.5 CPU 1212C ... 1457
A.5.1 Общие технические данные и характеристики ... 1457
A.5.2 Поддерживаемые CPU 1212C таймеры, счётчики и блоки кода ... 1459
A.5.3 Цифровые входы и выходы ... 1463
A.5.4 Аналоговые входы ... 1465
A.5.4.1 Реакция на скачок встроенных аналоговых входов CPU ... 1466
A.5.4.2 Время выборки для встроенных аналоговых портов CPU ... 1466
A.5.4.3 Диапазоны измерений для аналоговых входов напряжения (CPU) ... 1466
A.5.5 Схемы электрических соединений CPU 1212C ... 1467

A.6 CPU 1214C ... 1470
A.6.1 Общие технические данные и характеристики ... 1470
A.6.2 Поддерживаемые CPU 1214C таймеры, счётчики и блоки кода ... 1472
A.6.3 Цифровые входы и выходы ... 1476
A.6.4 Аналоговые входы ... 1478
A.6.4.1 Реакция на скачок встроенных аналоговых входов CPU ... 1479
A.6.4.2 Время выборки для встроенных аналоговых портов CPU ... 1479
A.6.4.3 Диапазоны измерений для аналоговых входов напряжения (CPU) ... 1479
A.6.5 Схемы электрических соединений CPU 1214C ... 1480

A.7 CPU 1215C ... 1484
A.7.1 Общие технические данные и характеристики ... 1484
A.7.2 Поддерживаемые CPU 1215C таймеры, счётчики и блоки кода ... 1486
A.7.3 Цифровые входы и выходы ... 1490
A.7.4 Аналоговые входы и выходы ... 1492
A.7.4.1 Реакция на скачок встроенных аналоговых входов CPU ... 1493
A.7.4.2 Время выборки для встроенных аналоговых портов CPU ... 1493
A.7.4.3 Диапазоны измерений для аналоговых входов напряжения (CPU) ... 1493
A.7.4.4 Технические данные аналоговых выходов ... 1494
A.7.5 Схемы электрических соединений CPU 1215C ... 1495

A.8 CPU 1217C ... 1499
A.8.1 Общие технические данные и характеристики ... 1499
A.8.2 Поддерживаемые CPU 1217C таймеры, счётчики и блоки кода ... 1501
A.8.3 Цифровые входы и выходы ... 1505
A.8.4 Аналоговые входы и выходы ... 1509
A.8.4.1 Технические данные аналоговых входов ... 1509
A.8.4.2 Реакция на скачок встроенных аналоговых входов CPU ... 1510

A.8.4.3 Время выборки для встроенных аналоговых портов CPU ... 1510
A.8.4.4 Диапазоны измерений для аналоговых входов напряжения (CPU) ... 1510
A.8.4.5 Технические данные аналоговых выходов ... 1511
A.8.5 Схемы электрических соединений CPU 1217C ... 1512
A.8.6 Подробные сведения и примеры использования дифференциального входа (DI) для CPU 1217C ... 1514
A.8.7 Подробные сведения и примеры использования дифференциального выхода (DQ) для CPU 1217C ... 1516

A.9 Цифровые сигнальные модули (SM) ... 1517
A.9.1 Технические данные для модуля цифрового ввода SM 1221 ... 1517
A.9.2 Технические данные для модуля цифрового вывода SM 1222 с 8 выходами ... 1519
A.9.3 Технические данные для модуля цифрового вывода SM 1222 с 16 выходами ... 1521
A.9.4 Технические данные для модуля цифрового ввода/вывода постоянного тока SM 1223 ... 1527
A.9.5 Технические данные для модуля цифрового ввода/вывода переменного тока SM 1223 ... 1534

A.10 Аналоговые сигнальные модули (SM) ... 1537
A.10.1 Технические данные модуля аналогового ввода SM 1231 ... 1537
A.10.2 Технические данные модуля аналогового вывода SM 1232 ... 1541
A.10.3 Технические данные модуля аналогового ввода/вывода SM 1234 ... 1543
A.10.4 Реакция на скачок аналоговых входов ... 1547
A.10.5 Время выборки и время обновления для аналоговых входов ... 1547
A.10.6 Диапазоны измерений аналоговых входов для напряжения и тока (SB и SM) ... 1548
A.10.7 Диапазоны измерений аналоговых выходов для напряжения и тока (SB и SM) ... 1549

A.11 Сигнальные модули для термопар и термосопротивлений (SM) ... 1550
A.11.1 SM 1231 Thermocouple (TC) ... 1550
A.11.1.1 Принцип работы термопары ... 1553
A.11.1.2 Таблицы выбора для SM 1231 Thermocouple (TC) ... 1554
A.11.2 SM 1231 RTD ... 1557
A.11.2.1 Таблицы выбора для SM 1231 RTD ... 1560

A.12 Технологические модули ... 1563
A.12.1 SM 1278 4xIO-Link-Master SM ... 1563
A.12.1.1 Обзор модуля SM 1278 4xIO-Link-Master ... 1566
A.12.1.2 Подключение ... 1569
A.12.1.3 Параметры/адресное пространство ... 1571
A.12.1.4 Прерывания, сообщения об ошибках и системные события ... 1574

A.13 Цифровые сигнальные платы (SB) ... 1577
A.13.1 Технические данные сигнальной платы цифрового ввода SB 1221 200 кГц ... 1577
A.13.2 Технические данные сигнальной платы цифрового вывода SB 1222 200 кГц ... 1580
A.13.3 Технические данные сигнальной платы цифрового ввода/вывода SB 1223 200 кГц ... 1583
A.13.4 Технические данные для SB 1223 с 2 x 24 В DC вход / 2 x 24 В DC выход ... 1587

A.14 Аналоговые сигнальные платы (SB) ... 1589
A.14.1 Технические данные сигнальной платы SB 1231 с 1 аналоговым входом ... 1589
A.14.2 Технические данные сигнальной платы SB 1232 с 1 аналоговым выходом ... 1592
A.14.3 Диапазоны измерения аналоговых входов и выходов ... 1594
A.14.3.1 Реакция на скачок аналоговых входов ... 1594
A.14.3.2 Время выборки и время обновления для аналоговых входов ... 1594
A.14.3.3 Диапазоны измерений аналоговых входов для напряжения и тока (SB и SM) ... 1594
A.14.3.4 Диапазоны измерений аналоговых выходов для напряжения и тока (SB и SM) ... 1596

A.14.4 Сигнальные платы для термопар (SB) ... 1597
A.14.4.1 Технические данные сигнальной платы SB 1231 с 1 аналоговым входом для термопары ... 1597
A.14.4.2 Принцип работы термопары ... 1599
A.14.5 Сигнальные платы RTD (SB) ... 1602
A.14.5.1 Технические данные сигнальной платы SB 1231 с 1 аналоговым входом RTD ... 1602
A.14.5.2 Таблицы выбора для SB 1231 RTD ... 1605
A.15 Плата буферной батареи BB 1297 ... 1607
A.16 Коммуникационные интерфейсы ... 1609
A.16.1 PROFIBUS ... 1609
A.16.1.1 CM 1242-5 ВЕДОМОЕ УСТРОЙСТВО PROFIBUS DP ... 1609
A.16.1.2 Разводка контактов для соединителя D-sub модуля CM 1242-5 ... 1610
A.16.1.3 CM 1243-5 ВЕДУЩЕЕ УСТРОЙСТВО PROFIBUS DP ... 1611
A.16.1.4 Разводка контактов для соединителя D-sub модуля CM 1243-5 ... 1612
A.16.2 CP 1242-7 ... 1613
A.16.2.1 CP 1242-7 GPRS ... 1613
A.16.2.2 GSM/GPRS антенна ANT794-4MR ... 1615
A.16.2.3 Плоская антенна ANT794-3M ... 1616
A.16.3 Коммуникационный модуль ведущего устройства AS-Interface CM 1243-2 ... 1616
A.16.3.1 Технические данные ведущего AS-i устройства CM 1243-2 ... 1616
A.16.3.2 Электрическое подключение ведущего устройства AS-i шины ... 1617
A.16.4 RS232, RS422 и RS485 ... 1619
A.16.4.1 Технические данные платы CB 1241 RS485 ... 1619
A.16.4.2 Технические данные модуля CM 1241 RS232 ... 1621
A.16.4.3 Технические данные модуля CM 1241 RS422/485 ... 1622
A.17 Телесервис (TS адаптер и модульный TS адаптер) ... 1624
A.18 Карты памяти SIMATIC ... 1624
A.19 Имитаторы входных сигналов ... 1625
A.20 Модуль потенциометров для S7-1200 ... 1626
A.21 Кабель для модулей расширения ... 1628
A.22 Сопутствующие продукты ... 1628
A.22.1 Блок питания PM 1207 ... 1628
A.22.2 Компактный коммутационный модуль CSM 1277 ... 1628
A.22.3 Коммуникационный модуль CM CANopen ... 1629
A.22.4 Коммуникационный модуль RF120C ... 1630
A.22.5 SM 1238 Energy Meter ... 1630
A.22.6 Электронные системы взвешивания SIWAREX ... 1630
B Расчет баланса мощностей ... 1633
C Информация для оформления заказа ... 1637
C.1 Модули CPU ... 1637
C.2 Сигнальные модули (SM), сигнальные платы (SB) и батарейные платы (BB) ... 1637
C.3 Коммуникация ... 1639
C.4 Оказобезопасные CPU и сигнальные модули ... 1640
C.5 Прочие модули ... 1641

C.6 Карты памяти 1641
C.7 Базовые устройства HMI 1641
C.8 Запасные части и другое оборудование 1642
C.9 ПО для программирования 1653
C.10 Лицензии OPC UA 1653
D Замена устройств и совместимость по запасным частям 1655
D.1 Замена CPU V3.0 на CPU V4.x 1655
D.2 S7-1200 до V3.0 - Запасные терминальные блоки 1661
Указатель 1663

Обзор продукта

1

1.1 Знакомство с контроллером S7-1200

Контроллер S7-1200 обладает необходимой гибкостью и мощностью для управления широким спектром задач автоматизации. Благодаря компактной конструкции, гибкой конфигурации и эффективному набору команд, контроллер S7-1200 является великолепным выбором для широкого спектра приложений.

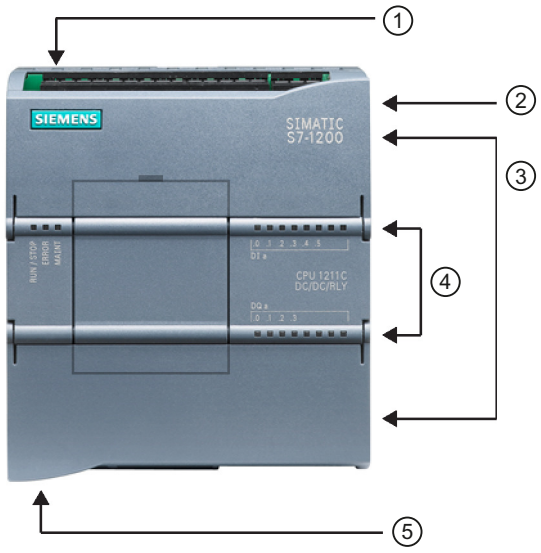
Модуль CPU объединяет следующие основные элементы в компактном корпусе, формируя мощную систему управления:

- микропроцессор
- встроенный блок питания
- входные и выходные цепи
- встроенный PROFINET
- высокоскоростные входы/выходы для управления перемещением

После загрузки программы, модуль CPU получает необходимую логику для контроля и управления устройствами в приложении. Модуль CPU выполняет мониторинг входов и изменяет выходы на основе команд программы пользователя, которая может включаться в себя логические операции, функции счета и времени, сложные арифметические операции, управление перемещением и коммуникацию с другими интеллектуальными устройствами.

Модуль CPU оснащен портом PROFINET для коммуникации с использованием сети PROFINET. Для коммуникации с использованием следующих сетей и протоколов предлагаются дополнительные модули:

- PROFIBUS
- GPRS
- LTE
- WAN со встроенными функциями безопасности (межсетевой экран, VPN)
- RS485
- RS232
- RS422
- IEC 60870
- DNP3
- USS
- MODBUS



① Разъем питания
② Слот для карты памяти под верхней крышкой
③ Съемный терминальный блок для подключений пользователя (под крышками)
④ Индикаторы состояния для встроенных I/O
⑤ Соединение PROFINET (на нижней стороне CPU)

Различные функции безопасности защищают доступ к модулю CPU и программе контроллера:

- Каждый CPU имеет парольную защиту (Страница 203) для гибкого управления доступом к функциям CPU.
- С помощью защиты ноу-хау (Страница 206) можно скрыть код в определенном блоке.
- С помощью защиты от копирования (Страница 207) можно привязать свою программу к определенной карте памяти или CPU.

Таблица 1-1 Сравнение модулей CPU

Особенность		**CPU 1211C**	**CPU 1212C**	**CPU 1214C**	**CPU 1215C**	**CPU 1217C**
Размеры (мм)		90 x 100 x 75		110 x 100 x 75	130 x 100 x 75	150 x 100 x 75
Память пользователя	Рабочая память	50 кБ	75 кБ	100 кБ	125 кБ	150 кБ
	Загружаемая память	1 МБ	2 МБ	4 МБ		
	Сохраняющая	10 кБ				
Локальные встроенные I/O	Цифровые	6 входов/ 4 выхода	8 входов/ 6 выходов	14 входов/ 10 выходов		
	Аналоговые	2 входа			2 входа/2 выхода	
Размер образа процесса	Входы (I)	1024 байт				
	Выход (Q)	1024 байт				
Меркер (M)		4096 байт		8192 байт		
Расширение: сигнальный модуль (SM)		Нет	2	8		
Сигнальная плата (SB), плата буферной батареи (BB) или коммуникационная плата (CB)		1				

Особенность		**CPU 1211C**	**CPU 1212C**	**CPU 1214C**	**CPU 1215C**	**CPU 1217C**
Коммуникационный модуль (CM) (установка слева)		3				
Быстрые счетчики	Всего	До 6 для использования встроенных или SB входов				
	1 МГц	-				Ib.2 до Ib.5
	100/[1]80 кГц	Ia.0 до Ia.5				
	30/[1]20 кГц	--	Ia.6 до Ia.7	Ia.6 до Ib.5		Ia.6 до Ib.1
	200 кГц[3]					
Импульсные выходы [2]	Всего	До 4 для использования встроенных или SB входов				
	1 МГц	--				Qa.0 до Qa.3
	100 кГц	Qa.0 до Qa.3				Qa.4 до Qb.1
	20 кГц	--	Qa.4 до Qa.5	Qa.4 до Qb.1		--
Карта памяти		Карта памяти SIMATIC (опция)				
Архивы данных	Количество	До 8 одновременно открытых				
	Размер	500 МБ на архив данных, или согласно ограничению по максимально доступной загружаемой памяти				
Буферизация часов реального времени		20 дней, тип./мин. 12 дней при 40 градусах C (необслуживаемый суперконденсатор)				
PROFINET Коммуникационный порт Ethernet		1			2	
Время выполнения арифм. операций		2,3 мкс/операция				
Время выполнения логических операций		0,08 мкс/операция				

[1] Более низкая скорость, если HSC сконфигурирован для режима счёта A/B.

[2] Для вариантов CPU с релейными выходами для использования импульсных выходов должна быть установлена цифровая сигнальная плата (SB).

[3] До 200 кГц доступно с SB 1221 DI x 24 В DC 200 кГц и SB 1221 DI 4 x 5 В DC 200 кГц.

Разные модификации CPU предлагают множество возможностей и функций для создания эффективных решений в различных областях применения. Подробную информацию по определенным модулям CPU можно найти в Технических данных (Страница 1433).

Таблица 1-2 Поддерживаемые S7-1200 блоки кода, таймеры и счётчики

Элемент			**Описание**				
Блоки	Тип		OB, FB, FC, DB				
	Раз-мер	**Вариант CPU**	**CPU 1211C**	**CPU 1212C**	**CPU 1214C**	**CPU 1215C**	**CPU 1217C**
		Блоки кода	50 кБ	64 кБ	64 кБ	64 кБ	64 кБ
		Связанные[1] блоки данных	50 кБ	75 кБ	100 кБ	125 кБ	150 кБ
		Несвязанные[2] блоки данных	256 кБ	256 кБ	256 кБ	256 кБ	256 кБ
	Количество		В сумме до 1024 блоков (OB + FB + FC + DB)				
	Глубина вложенности		16 из OB циклов или пусковых OB 6 из любых OB прерывающих событий[3]				
	Контроль		Контроль состояния 2 блоков кода одновременно.				
OB	Цикл программы		Несколько				
	Пуск		Несколько				
	Прерывания по задержке времени		4 (1 на событие)				
	Циклические прерывания		4 (1 на событие)				
	Аппаратные прерывания		50 (1 на событие)				
	Прерывания по ошибке времени		1				
	Диагностические прерывания		1				
	Извлечение или вставка модулей		1				
	Ошибка стойки или станции		1				
	Прерывание по времени		Несколько				
	Состояние		1				
	Обновление		1				
	Профиль		1				
Таймеры	Тип		IEC				
	Количество		Ограничивается только объемом памяти				
	Сохранение		Структура в DB, 16 байт на таймер				
Счетчик	Тип		IEC				
	Количество		Ограничивается только объемом памяти				
	Сохранение		Структура в DB, размер зависит от типа счёта • SInt, USInt: 3 байта • Int, UInt 6 байт • DInt, UDInt: 12 байт				

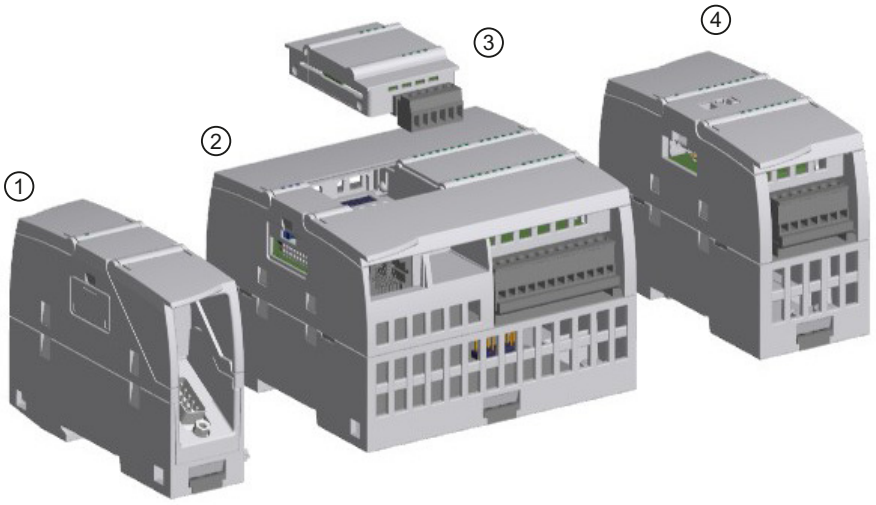
[1] Сохранение в рабочей и загружаемой памяти. Без превышения размера свободной рабочей или загружаемой памяти.

[2] Сохраняется только в загружаемой памяти

[3] Программы обеспечения безопасности используют два уровня вложенности. Таким образом, программа пользователя имеет в программах обеспечения безопасности глубину вложенности в четыре уровня.

1.2 Возможности расширения CPU

Семейство продуктов S7-1200 предлагает множество модулей и съемных плат для расширения возможностей CPU путем добавления дополнительных входов-выходов или других коммуникационных протоколов. Подробную информацию по определенным модулям CPU можно найти в Технических данных (Страница 1433).



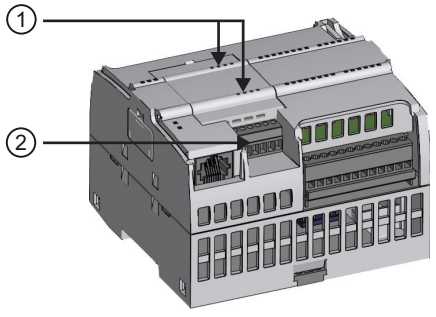
① Коммуникационный модуль (CM) или коммуникационный процессор (CP) (Страница 1609)

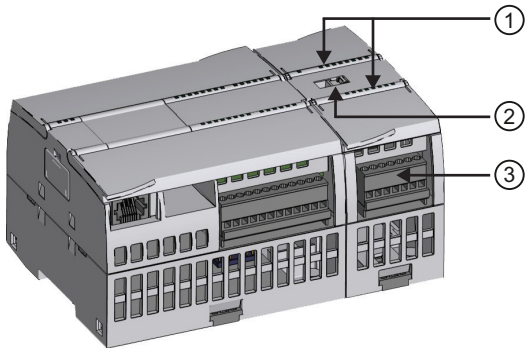
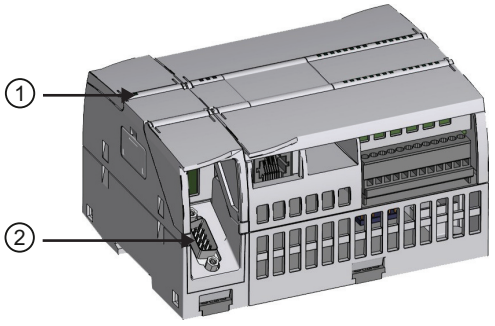
② CPU (CPU 1211C (Страница 1444), CPU 1212C (Страница 1457), CPU 1214C (Страница 1470), CPU 1215C (Страница 1484), CPU 1217C (Страница 1499))

③ Сигнальная плата (SB) (цифровая SB (Страница 1577), аналоговая SB (Страница 1589)), коммуникационная плата (CB) (Страница 1619) или плата буферной батареи (BB) CPU (CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C, CPU 1217C) (Страница 1607)

④ Сигнальный модуль (SM) (цифровой SM (Страница 1517), аналоговый SM (Страница 1537), модуль измерения температуры SM TC (Страница 1550), SM RTD (Страница 1557), технологический SM) (Страница 1563)

Таблица 1-3 S7-1200 модули расширения

Тип модуля	**Описание**
Модуль CPU поддерживает одну съемную плату расширения: • Сигнальная плата (SB) содержит дополнительные входы/выходы для CPU. SB подключается с лицевой стороны CPU. • С помощью коммуникационной платы (CB) можно добавить порт передачи данных к CPU. • Плата буферной батареи (BB) обеспечивает долговременную буферизацию часов реального времени.	 ① Индикаторы состояния на SB ② Съемный терминальный блок для подключений пользователя

Тип модуля	Описание
Сигнальные модули (SM) расширяют функциональность модулей CPU. SM подключаются с правой стороны модуля CPU. • Цифровые I/O • Аналоговые I/O • RTD и TC • SM 1278 IO-Link-Master • SM 1238 Energy Meter (https://support.industry.siemens.com/cs/ww/de/view/109483435)	 ① Индикаторы состояния ② Ползунок-фиксатор шинного соединителя ③ Съемный терминальный блок для подключений пользователя
Коммуникационные модули (CM) и коммуникационные процессоры (CP) добавляют к CPU различные возможности для передачи данных, напр., B. PROFIBUS или RS232/RS485 (для PtP, Modbus или USS) или AS-i-Master. Коммуникационный процессор CP предлагает возможности для других способов коммуникации, напр., для подключения модуля CPU с использованием сетей GPRS, LTE, IEC, DNP3 или WDC. • К одному CPU может быть подключено до 3 CM или CP. • Каждый CM или каждый CP подключается с левой стороны модуля CPU (или с левой стороны другого CM или CP).	 ① Индикаторы состояния ② Коммуникационный коннектор

1.3 Базовые панели HMI

Базовые панели SIMATIC HMI - это устройства с сенсорным экраном для решения базовых задач в области оперативного управления и мониторинга. Все панели имеют класс защиты IP65 и сертифицированы по CE, UL, cULus и NEMA 4x.

Доступные базовые панели HMI (Страница 1641)представлены ниже:

- KTP400 Basic: 4" сенсорный экран с 4 конфигурируемыми клавишами, разрешение 480 x 272 и 800 переменных (тегов)
- KTP700 Basic: 7" сенсорный экран с 8 конфигурируемыми клавишами, разрешение 800 x 480 и 800 переменных (тегов)
- KTP700 Basic DP: 7" сенсорный экран с 8 конфигурируемыми клавишами, разрешение 800 x 480 и 800 переменных (тегов)

- KTP900 Basic: 9" сенсорный экран с 8 конфигурируемыми клавишами, разрешение 800 x 480 и 800 переменных (тегов)
- KTP1200 Basic: 12" сенсорный экран с 10 конфигурируемыми клавишами, разрешение 800 x 480 и 800 переменных (тегов)
- KTP 1200 Basic DP: 12" сенсорный экран с 10 конфигурируемыми клавишами, разрешение 800 x 400 и 800 переменных (тегов)

Новые функции

2

Следующие функции появились в версии 4.4 впервые:

- Сервер OPC UA (Страница 1017) - S7-1200 CPU поддерживают часть функций OPC UA
- Обновление открытых коммуникационных соединений пользователя (Страница 828):
  - Поддержка электронных писем с прикрепленными файлами пользователя (рецепты и архивы данных) через TMAIL_C
  - Разрешение имен DNS через TMAIL_C
  - Поддержка "InterfaceId 0", когда модуль CPU выбирает соответствующий интерфейс CPU (не для модулей CP).
  - Поддержка разрешения имен DNS для коммуникации TCP/UDP
- Обновление инструкций:
  - SCATTER, SCATTER_BLK, GATHER и GATHER_BLK (Страница 279)
- Управление перемещением: Инструкция MC_Reset (подтвердить ошибку) может квитировать ошибки в очереди до активации оси программой пользователя.
- Веб-сервер
  - Непрерывная гармонизация стандартных веб-страниц между S7-1200 и S7-1500
  - Поддержка для обновления прошивки сконфигурированных PROFINET IO устройств или модулей
  - С помощью загрузки архива данных / удаления загрузки архива можно:
    - Отобразить список всех архивов данных на PLC
    - Выгрузить архив данных из PLC в компьютер
    - Удалить архив данных из PLC
    - Вызвать и удалить архив данных из PLC
  - Браузер файлов пользователя
- Два новых цифровых сигнальных модуля предлагают выходы со снижением тока (выходная нагрузка с негативной логикой (тип M)):
  - SM 1223 DI 16 x 24 В DC, DQ 16 x 24 В DC со снижением тока (6ES7223-1BL32-1XB0) (Страница 1527)
  - SM 1222 DQ 16 x 24 В DC со снижением тока (6ES7222-1BH32-1BX0) (Страница 1521)

Замена CPU V3.0 на CPU V4.x.x

При замене S7-1200 CPU V3.0 на S7-1200 CPU V4.x.x следует учитывать задокументированные различия (Страница 1655) между версиями и необходимые мероприятия.

См. также

Параметры для соединения PROFINET (Страница 835)

S7-1200 руководство по функциональной безопасности (https://support.industry.siemens.com/cs/ww/en/view/104547552)

ПО для программирования STEP 7

3

ПО STEP 7 предлагает эргономичное окружение для разработки, редактирования и мониторинга логических структур, необходимых для управления приложением пользователя. Также оно содержит инструментарий для конфигурирования всех устройств в проекте, напр., устройств PLC и HMI. Для поиска необходимой информации в STEP 7 есть обширная система интерактивной помощи.

STEP 7 предлагает стандартные языки программирования для удобной и эффективной разработки программы управления для приложения.

- LAD (релейно-контактные схемы) (Страница 192) - это графический язык программирования. Для отображения используются принципиальные схемы.
- FBD (функциональная блок-схема) (Страница 193) - это язык программирования на основе графических логических символов Булевой алгебры.
- SCL (структурированный язык управления) (Страница 194) - это язык программирования высокого уровня на основе текстов.

При создании блока кода следует выбрать язык программирования для блока. Программа пользователя может работать с блоками кода, созданными с помощью одного из этих языков программирования.

Примечание

STEP 7 это программный компонент для программирования и конфигурирования в TIA-Portal. TIA-Portal, наряду с STEP 7, содержит и WinCC для разработки и выполнения рабочей визуализации процессов, а также предлагает систему интерактивной помощи для WinCC, как и для STEP 7.

Для новых функций в S7-1200 V4.4 потребуется STEP 7 Professional V16.

3.1 Системные требования

Необходимо установить STEP 7 с правами администратора.

Таблица 3-1 Системные требования

Аппаратное / программное обеспечение	Необходимые условия
Тип процессора	Intel® Core™ i3-6100U, 2,30 ГГц или выше
RAM	8 ГБ
Свободное место на жестком диске	20 ГБ на системном диске "C:"

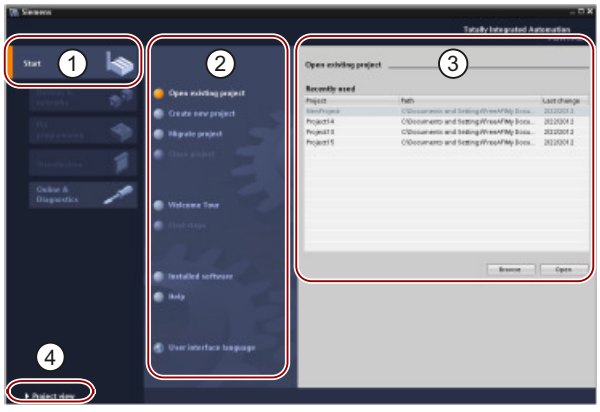
Аппаратное / программное обеспечение	**Необходимые условия**
Операционные системы	ПО STEP 7 может работать со следующими операционными системами: • Windows 7 (64 бита): – Windows 7 Home Premium SP1 ** – Windows 7 Professional SP1 – Windows 7 Enterprise SP1 – Windows 7 Ultimate SP1 • Windows 10 (64 бита): – Windows 10 Home Version 1709 ** – Windows 10 Home Version 1803 ** – Windows 10 Professional Version 1709 – Windows 10 Professional Version 1803 – Windows 10 Enterprise Version 1709 – Windows 10 Enterprise Version 1803 – Windows 10 Enterprise 2016 LTSB – Windows 10 IoT Enterprise 2015 LTSB – Windows 10 IoT Enterprise 2016 LTSB • Windows Server (64 бита) – Windows Server 2012 R2 StdE (полная установка) – Windows Server 2016 Standard (полная установка)
Графический адаптер	32 МБ RAM память 24 бит глубина цветопередачи
Разрешение экрана	1024 x 768
Сеть	100 Мбит/с Ethernet или более быстрый для обмена данными между STEP 7 и CPU

* Включая все применимые обновления безопасности. Дополнительные сведения об операционных системах приведены в справке Microsoft Windows или на домашней странице Microsoft.

** Только для базовых версий

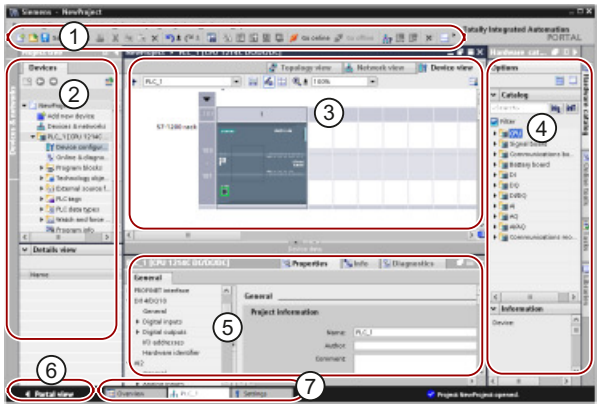
3.2 Простая работа с различными представлениями

STEP 7 предоставляет удобную в использовании рабочую среду для разработки логики управления, конфигурирования визуализации HMI и настройки сетевой коммуникации. Для увеличения производительности STEP 7 предлагает два различных представления проекта: ориентированный на выполнение задач набор порталов для отдельных функций (портальное представление) и ориентированное на проект представление элементов внутри проекта (проектное представление). Пользователь может выбрать представление для более эффективной организации своего процесса работы. Переключение между портальным и проектным представлением осуществляется кликом мыши.



Портальное представление

① Порталы для решения различных задач
② Задачи для выбранного портала
③ Панель выбора для отмеченной операции
④ Переход в проектное представление



Проектное представление

① Меню и панель инструментов
② Навигатор по проекту
③ Рабочая область
④ Окна задач
⑤ Окно инспектора
⑥ Переход в портальное представление
⑦ Панель редакторов

Так как все компоненты собраны в одном месте, пользователь имеет быстрый доступ к любой области своего проекта. Рабочая область состоит из трех вкладок:

- Просмотр устройств: Показывает устройство, которое было добавлено или выбрано, а также относящиеся к нему модули
- Просмотр сетевых соединений: Показывает модули CPU и сетевые соединения в сети пользователя
- Просмотр топологии: Показывает топологию PROFINET сети, включая устройства, пассивные компоненты, порты, соединения и диагностику портов

В каждой вкладке можно выполнять и задачи конфигурирования. Окно инспектора отображает свойства и другую информацию о выбранном в рабочей области объекте. Для различных выбранных пользователем объектов окно инспектора отображает соответствующие конфигурируемые свойства. Окно инспектора также предлагает вкладки с диагностической информацией и дополнительными сообщениями.

На панели редакторов отображаются все открытые в настоящее время редакторы. Панель редакторов обеспечивает быструю и эффективную работу. Для переключения на необходимый редактор следует просто кликнуть на нем. Также возможно одновременное отображение двух редакторов с их вертикальным или горизонтальным расположением. Операции "перетаскивания" между редакторами поддерживаются.

Информационная система STEP 7 предлагает обширную систему интерактивной помощи для всех инструментов конфигурирования, программирования и мониторинга STEP 7. Также она содержит подробные объяснения, выходящие за рамки данного справочника.

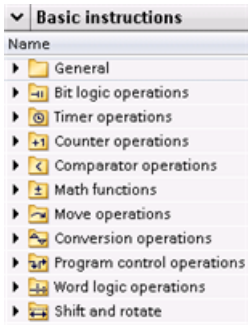
3.3 Эргономичные инструменты

3.3.1 Вставка инструкций в программе пользователя

STEP 7 предлагает окна задач с инструкциями для программы. Инструкции отсортированы по функциям.



Для создания программы необходимо перетащить инструкции из окна задач в сегмент программы.

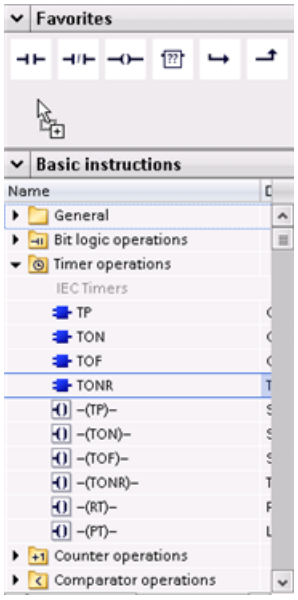


3.3.2 Доступ к инструкциям на панели инструментов "Избранное"

У STEP 7 есть панель инструментов "Избранное" для быстрого доступа к часто используемым инструкциям. Для добавления инструкции в требуемый сегмент программы необходимо лишь кликнуть по соответствующему символу.



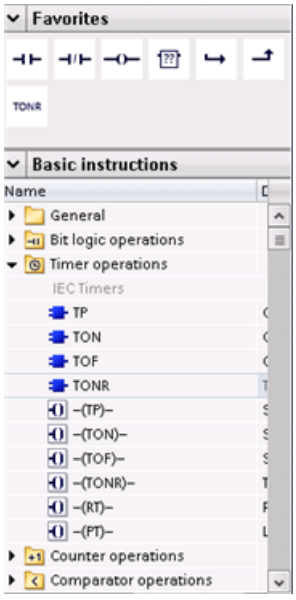
(Дважды кликнуть по символу для вызова "Избранного" в дереве инструкций.)



Можно легко настроить панель инструментов "Избранное" путем добавления новых инструкций.

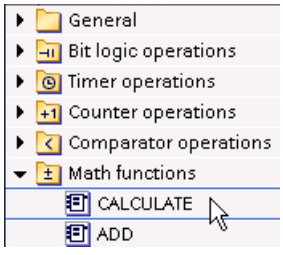
Просто перетащить необходимую инструкцию в "Избранное".

После этого инструкция доступна по клику на ней!

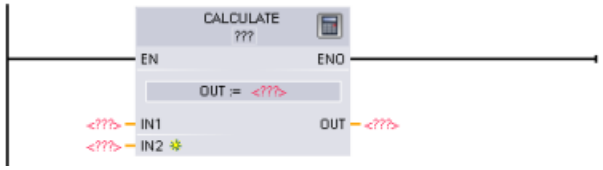


3.3.3 Создание сложного уравнения с помощью простой инструкции

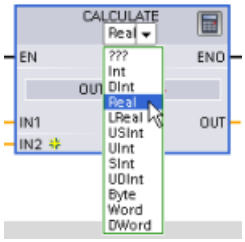
Инструкция Calculate (Страница 247) позволяет создать математическую функцию, которая производит действие над несколькими входными параметрами для получения результата согласно заданному пользователем уравнению.



Открыть папку математических функций в дереве базовых инструкций. Двойной клик по инструкции Calculate для вставки её в программу пользователя.



Несконфигурированная инструкция Calculate предлагает два входных и один выходной параметр.

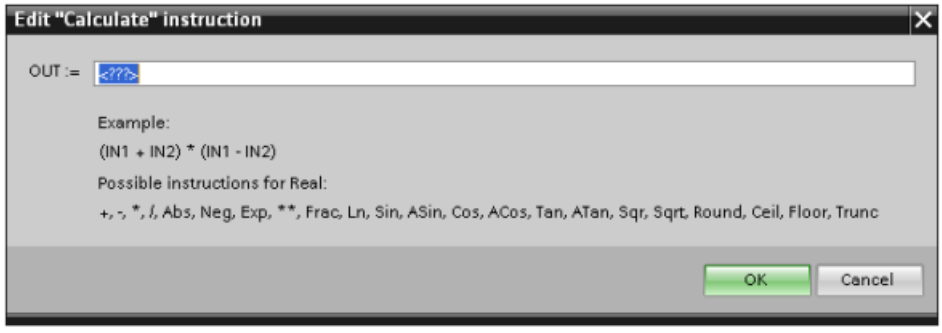


Кликнуть по "???" и выбрать типы данных для входных и выходных параметров. (Все входные и выходные параметры должны иметь один и тот же тип данных.)

Выбрать для данного примера тип данных "Real".



Кликнуть по символу "Редактировать уравнение", чтобы ввести уравнение.



Ввести для этого примера следующее уравнение для масштабирования необработанного аналогового значения. (Обозначения "In" и "Out" соответствуют параметрам инструкции Calculate.)

$Out_{value} = ((Out_{high} - Out_{low}) / (In_{high} - In_{low})) * (In_{value} - In_{low}) + Out_{low}$

Out = ((in4 - in5) / (in2 - in3)) * (in1 - in3) + in5

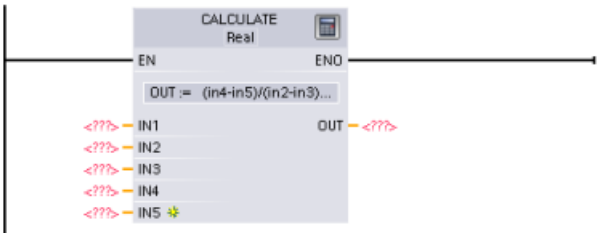
Объяснение:	Out_{value}	(Out)	Масштабированное выходное значение
	In_{value}	(in1)	Аналоговое входное значение
	In_{high}	(in2)	Верхняя граница для масштабированного входного значения
	In_{low}	(in3)	Нижняя граница для масштабированного входного значения
	Out_{high}	(in4)	Верхняя граница для масштабированного выходного значения
	Out_{low}	(in5)	Нижняя граница для масштабированного выходного значения

В окне "Редактировать Calculate" ввести уравнение с именами параметров:

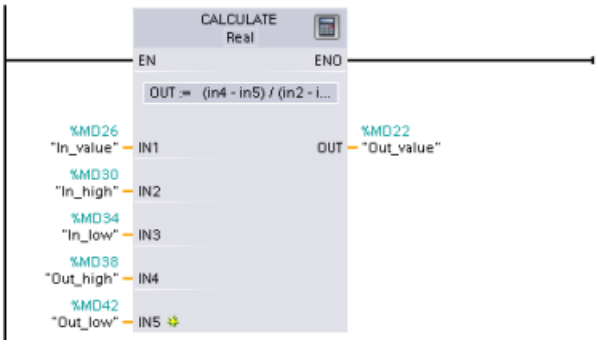
OUT = ((in4 - in5) / (in2 - in3)) * (in1 - in3) + in5



После нажатия на кнопку "OK" инструкция Calculate создает входы, необходимые для инструкции.



Ввести имена переменных (тегов) для значений, которые соответствуют параметрам.



3.3.4 Добавление входов или выходов в LAD или FBD инструкцию

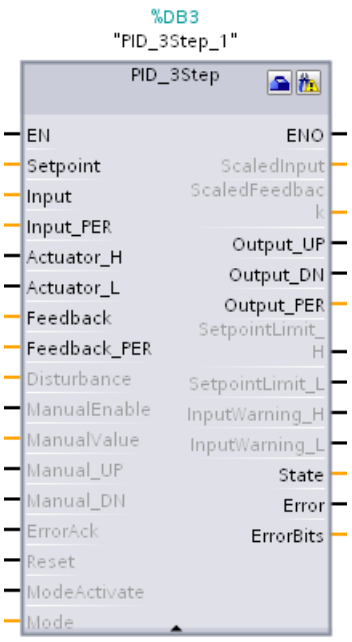


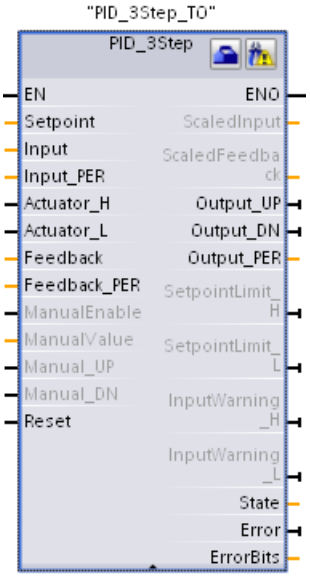
Некоторые инструкции позволяют создавать дополнительные входы или выходы.

- Для добавления входа или выхода кликнуть левой кнопкой мыши по символу "Создать" или правой кнопкой мыши на входном соединении одного из параметров IN или OUT и выбрать команду "Вставить вход".
- Для удаления входа или выхода кликнуть правой кнопкой мыши на соединении одного из параметров IN или OUT (если имеется более двух исходных входов) и выбрать команду "Удалить".

3.3.5 Расширяемые инструкции

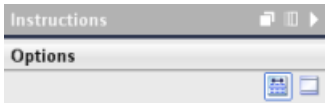
Некоторые из более сложных инструкций являются расширяемыми, отображая в начале только ключевые входы и выходы. Чтобы показать все входы и выходы, нажать на стрелку в нижней части инструкции.



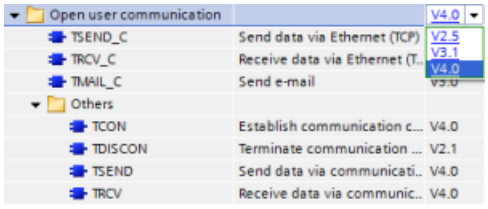


3.3.6 Выбор версии для инструкции

Циклы разработки и выпуска для определенных наборов инструкций (напр., Modbus, ПИД и перемещение) привели к наличию нескольких доступных версий для этих инструкций. Для обеспечения совместимости и миграции из старых проектов, STEP 7 позволяет выбирать версию инструкции, вставляемой в программу пользователя.



Кликнуть по символу в окне задач с деревом инструкций, чтобы активировать заголовки и столбцы в дереве инструкций.

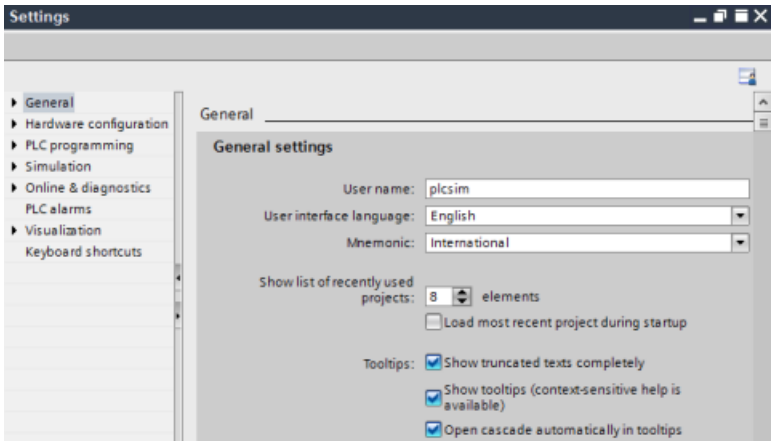


Для изменения версии инструкции, выбрать соответствующую версию в раскрывающемся списке.

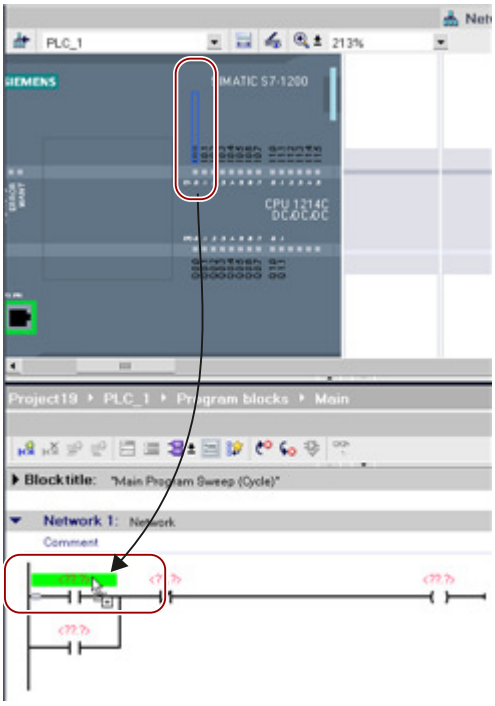
3.3.7 Изменение внешнего вида и конфигурации STEP 7

Предлагается множество настроек, например, для изменения внешнего вида интерфейса пользователя, языка или папки для сохранения результатов работы.

Для изменения настроек выбрать команду "Параметры" в меню "Опции".



3.3.8 Перетаскивание между редакторами

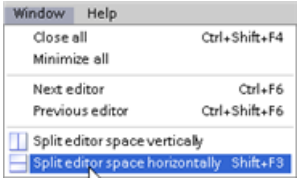


Для быстрого и эффективного решения задач, STEP 7 позволяет перетаскивать элементы из одного редактора в другой с помощью мыши. Например, можно перетащить вход CPU на адрес инструкции в программе пользователя.

Для выбора входов или выходов CPU, необходимо увеличить масштаб по крайней мере до 200%.

Следует учитывать, что имена переменных (тегов) отображаются не только в таблице переменных PLC, но и в CPU.

Для одновременного отображения двух редакторов следует использовать команду меню "Разделить редактор" или соответствующие кнопки на панели инструментов.





Для переключения между открытыми редакторами, кликать по соответствующим символам на панели редакторов.



3.3.9 Изменение рабочего состояния CPU

У CPU нет физического переключателя для изменения рабочего состояния (STOP или RUN).

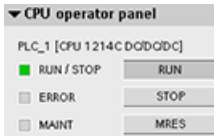
Использовать кнопки панели инструментов "Запустить CPU" или "Остановить CPU", чтобы изменить рабочее состояние CPU.



При настройке CPU в конфигурации устройства (Страница 151) задается его пусковая характеристика (Страница 167).

Кроме этого, портал "Онлайн и диагностика" предлагает панель управления для изменения рабочего состояния CPU в режиме онлайн. Для использования панели

управления CPU необходимо интерактивное соединение с CPU. Панель управления в окне задач "Интерактивные инструменты" показывает рабочее состояние интерактивного CPU. С помощью панели управления можно изменять рабочее состояние интерактивного CPU.



Изменить рабочее состояние с помощью экранной кнопки на панели управления (STOP или RUN). Кроме этого, на панели управления имеется экранная кнопка MRES для восстановления исходного состояния памяти.

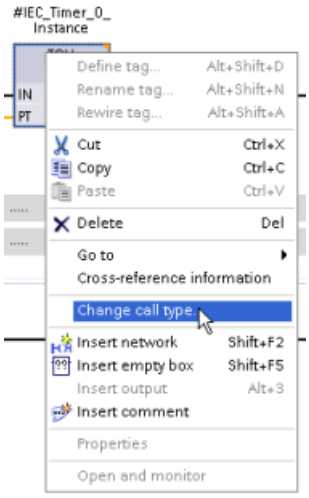
Цвет индикатора RUN/STOP показывает текущее рабочее состояние CPU. Желтый цвет указывает на состояние STOP, а зеленый - на состояние RUN.

В конфигурации устройства в STEP 7 (Страница 151) можно настроить рабочее состояние по умолчанию при включении CPU (Страница 87).

Примечание

Для изменения рабочего состояния CPU также можно использовать веб-сервер (Страница 1023) или SIMATIC Automation Tool (https://support.industry.siemens.com/cs/ww/en/view/98161300).

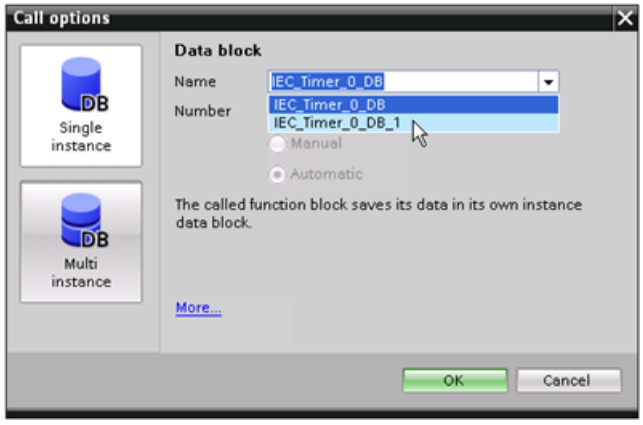
3.3.10 Изменение типа вызова DB



STEP 7 позволяет легко создать или изменить связь DB с инструкцией или FB, находящимся в другом FB.

- Можно выбирать связь с различными DB.
- Можно выбирать связь либо с экземплярным, либо с мультиэкземплярным DB.
- Можно создавать DB экземпляра (если DB экземпляра отсутствует или недоступен).

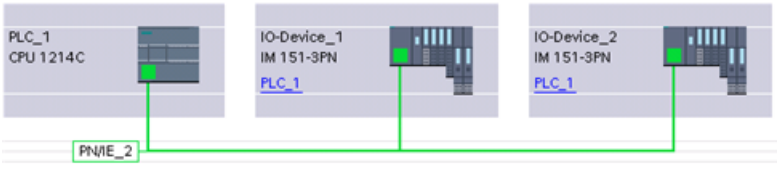
Можно выбрать команду "Изменить тип вызова" либо кликнув правой кнопкой мыши по инструкции или FB в редакторе программ, либо выбирая команду "Вызов блока" из меню "Опции".

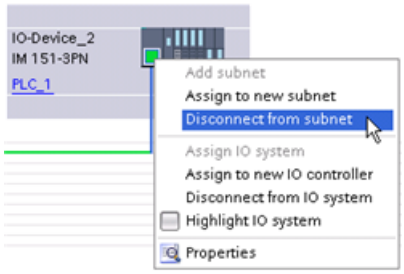


В диалоге "Опции вызова" можно выбрать экземплярный или мульти-экземплярный DB. Также можно выбирать определенные DB из раскрывающегося списка доступных DB.

3.3.11 Временное отключение устройств от сети

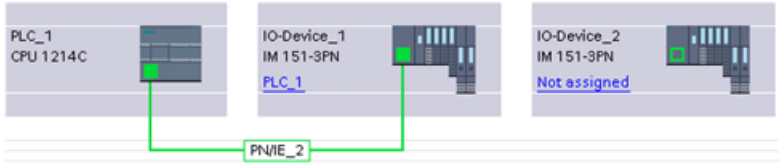
В виде сети можно отключить отдельные сетевые устройства от подсети. Поскольку конфигурация устройства не удаляется из проекта, можно легко восстановить соединение с устройством.





Кликнуть правой кнопкой мыши по интерфейсному порту сетевого устройства и выбрать команду "Отсоединить от подсети" из контекстного меню.

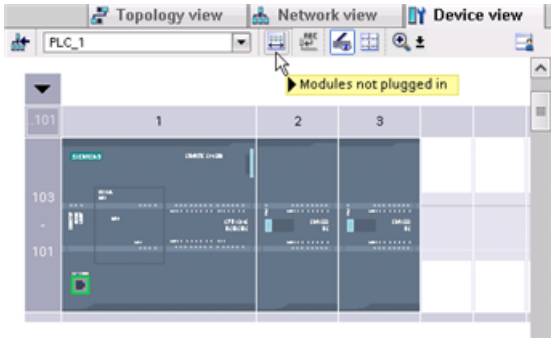
STEP 7 повторно конфигурирует сетевые соединения, но не удаляет отключенное устройство из проекта. Несмотря на то, что сетевое соединение удаляется, адреса интерфейсов не изменяются.

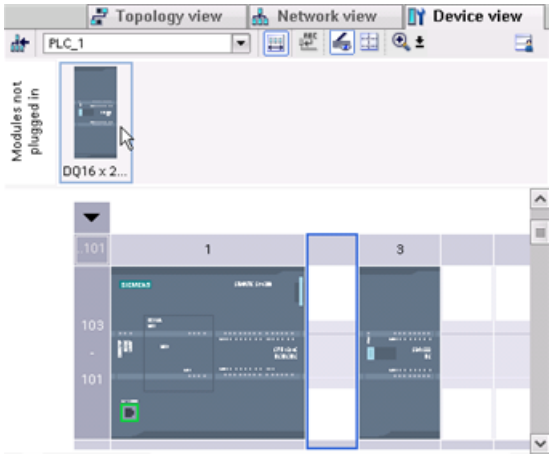


При загрузке новых сетевых соединений, CPU должен быть переведен в рабочее состояние STOP.

Для повторного подключения устройства следует просто создать новое сетевое соединение с портом устройства.

3.3.12 Виртуальное исключение устройств из конфигурации





STEP 7 предлагает место для хранения "отключенных" модулей. Можно "извлечь" модуль из стойки, сохранив при этом его конфигурацию. Эти отключенные модули сохраняются вместе с проектом, позволяя повторно вставить их в будущем без необходимости повторно конфигурировать параметры.

Одним из вариантов использования этой функции является срочное обслуживание. Представьте случай, когда должен поступить модуль для замены и планируете временно использовать другой модуль. В этом случае можно перетащить сконфигурированный модуль из стойки в "место хранения отключенных модулей" и после использовать временный запасной модуль.

3.4 Совместимость с предыдущими версиями

STEP 7 V16 поддерживает конфигурирование и программирование CPU S7-1200 V4.4.

Но можно загрузить проекты для более ранних версий S7-1200 CPU V4.x из STEP 7 V13 SP1 и выше в S7-1200 CPU V4.4. Конфигурация и программа будут ограничены набором функций и инструкций, которые поддерживаются прежней версией S7-1200 CPU и используемой версией STEP 7.

Такая совместимость с предыдущими версиями позволяет запускать на новых версиях S7-1200 V4.4 CPU программы, которые ранее разрабатывались и программировались для более старых версий.

⚠ ПРЕДУПРЕЖДЕНИЕ

Риски при копировании и вставке логики программы из старых версий STEP 7

Копирование логики программы из старой версии STEP 7 может стать причиной непредсказуемого поведения при выполнении программы или ошибок при компиляции. В разных версиях STEP 7 элементы программы реализуются по разному. При компиляции различия обнаруживаются не всегда, если изменения были внесены путем вставки из старой версии в STEP 7 V15. Выполнение непредсказуемой логики программы может стать причиной тяжких или опасных для жизни телесных повреждений, если программа не будет исправлена.

При использовании логики программы из старой версии STEP 7 всегда следует обновлять весь проект до последней версии STEP 7. При необходимости можно копировать, вырезать, вставлять и редактировать логику программы. В STEP 7 V16 можно открыть проект из STEP 7 V13 SP1 или выше. После STEP 7 выполняет необходимые для обеспечения совместимости преобразования и правильно обновляет программу. Такие преобразования и исправления в комбинации с обновлением необходимы для правильной компиляции и выполнения программы. Если проект старше, чем STEP 7 V13 SP1, то следует пошагово обновить проект до STEP 7 V16 (Страница 1655).

Нельзя загрузить проекты S7-1200 CPU V1.0, V2.0 или V3.0 в S7-1200 CPU V4.x. В Замена устройств и совместимость по запасным частям (Страница 1655) можно найти информацию о обновлении старых проектов до версии, которая может быть загружена в целевую систему.

Примечание

Проекты с версиями S7-1200 CPU V1.x

Нельзя открыть проект STEP 7, содержащий S7-1200 CPU V1.x, в STEP 7 V15.1. Для использования существующего проекта, он должен быть открыт в STEP 7 V13 SP1 (с любым обновлением) с последующим преобразованием S7-1200 CPU V1.x в V2.0 или выше. После можно открыть сохраненный проект с преобразованными CPU в STEP 7 V15.1.

См. также

Новые функции (Страница 35)

Монтаж

4

4.1 Руководства по монтажу устройств S71200

Конструкция устройств S7-1200 обеспечивает простой монтаж. Система S7-1200 может быть установлена на пластину или на стандартную DIN–рейку; разрешается монтаж в горизонтальном или вертикальном положении. Компактные размеры S7-1200 позволяют эффективно использовать пространство.

Стандарты на электрооборудование классифицируют систему SIMATIC S7-1200 как "Открытое оборудование". Система S7-1200 должна устанавливаться в корпусе, шкафу или электрощитовой. Только уполномоченный персонал может иметь доступ к корпусу, шкафу или электрощитовой.

Установка должна обеспечить сухую окружающую среду для S7-1200. Цепи тока SELV/ PELV обеспечивают защиту от поражения током в сухих помещениях.

Установка должна обеспечить необходимую механическую устойчивость, пожаробезопасность и стабильность, определенные для открытого оборудования в конкретной категории размещения в действующих электротехнических и строительных нормах и правилах.

Попадание пыли, влаги и частиц из загрязненного воздуха на проводящие части могу вызвать эксплуатационные и электрические отказы в PLC.

Если PLC размещается в месте, где возможно загрязнение электропроводящих частей, следует поместить его в корпус с соответствующим классом защиты. IP54 - это класс защиты, который обычно используется для электронного оборудования в сильно загрязненной окружающей среде и, возможно, подходит для конкретного случая использования.

⚠ ПРЕДУПРЕЖДЕНИЕ
Неправильная установка S7-1200 может привести к электрическим отказам или непредсказуемой работе оборудования.
Электрические отказы или непредсказуемая работа оборудования могут стать причиной тяжких или опасных для жизни травм и/или материального ущерба.
Для обеспечения безопасной эксплуатации оборудования необходимо следовать всем инструкциям по организации и обслуживанию надлежащего рабочего окружения.

Устройства S7-1200 не должны подвергаться воздействию тепла, высоких напряжений или электрических помех

Общее правило для размещения устройств в системе состоит в том, что устройства, генерирующие высокое напряжения или сильные электрические помехи, должны быть удалены от низковольтных электронных приборов, таких, как S7-1200.

При размещении S7-1200 на пластину следует обратить внимание на приборы, выделяющие тепло, и размещать электронные устройства в более прохладных местах

шкафа. Чем меньше электронное устройство находится в высокотемпературной среде, тем больше срок его службы.

Также следует обратить внимание на прокладку проводки для устройств на пластине. Избегать прокладки низковольтных сигнальных кабелей и коммуникационных кабелей в одном канале с кабелями питания переменного тока и часто коммутируемыми силовыми кабелями постоянного тока.

Достаточное место для охлаждения и проводки

Устройства S7-1200 предусматриваю естественное охлаждение путем конвекции. Поэтому для надлежащего отвода тепла необходимо предусмотреть отступ не менее 25 мм над и под устройством. Расстояние от лицевой стороны модуля до внутренней стороны корпуса должно быть не менее 25 мм.

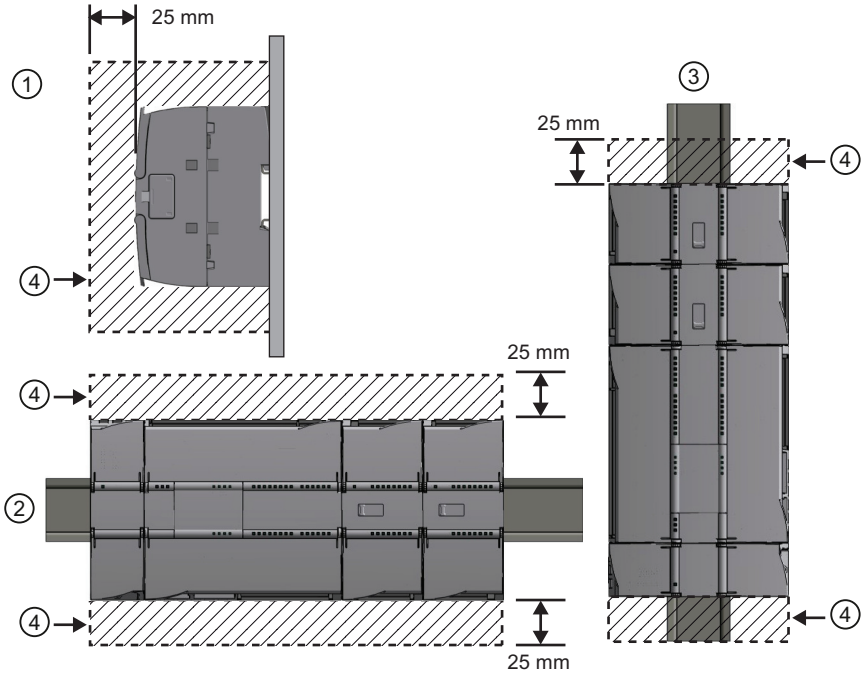
⚠ ОСТОРОЖНО

Для вертикальной установки максимальная допустимая температура окружающей среды ниже на 10 °C.

Точная установка вертикально смонтированной системы S71200 показано на рисунке ниже.

Убедиться, что система S7-1200 смонтирована правильно.

При планировании размещения системы S7-1200 следует оставить достаточно места для проводки и подключений коммуникационного кабеля.



① Вид сбоку
② Горизонтальная монтажная позиция
③ Вертикальная монтажная позиция
④ Свободное пространство

4.2 Баланс мощности

В CPU имеется внутренний источник питания 24 В DC, который, наряду с CPU, обеспечивает энергией сигнальные модули/платы, коммуникационные модули и другие нагрузки.

В технических данных (Страница 1433) можно найти информацию о балансе мощности для напряжения 5 В пост. тока для CPU и потребляемой мощности 5 В DC для сигнальных модулей, сигнальных плат и коммуникационных модулей. С помощью информации в разделе "Расчет баланса мощности" (Страница 1633) можно рассчитать, какую мощность (или ток) модуль CPU может предоставить для конкретной конфигурации.

Модуль CPU также имеет блок питания датчиков 24 В DC, который обеспечивает питание 24 В DC для входов, катушек реле сигнальных модулей и других нагрузок. Если потребляемая мощность 24 В DC превышает возможности источника питания датчиков, то следует подключить к системе внешний блок питания 24 В DC. В технических данных (Страница 1433) можно узнать баланс мощности для источника питания датчиков 24 В DC для соответствующего CPU.

Если используется внешний источник питания 24 В DC, то следует проследить, чтобы он не был подключен параллельно с источником питания датчиков CPU. Для достижения наилучшей стойкости к воздействию помех рекомендуется соединить между собой кабели (M) соответствующих источников питания.

⚠ ПРЕДУПРЕЖДЕНИЕ

Подключение внешнего источника питания 24 В DC параллельно источнику питания датчиков 24 В DC может привести к конфликту между двумя источниками, так как каждый из них стремится установить свой собственный уровень выходного напряжения.

Результатом этого конфликта может быть сокращение срока службы или немедленный выход из строя одного или обоих источников питания. Непредсказуемое поведение системы может привести к тяжким телесным повреждениям и/или материальному ущербу.

Блок питания датчиков постоянного тока и внешний источник питания должны подключаться к различным нагрузкам.

Некоторые входные порты 24 В DC системы S7-1200 соединены друг с другом, при этом общий логический нулевой провод соединяет между собой несколько клемм M. Например, следующие цепи соединены друг с другом, если в технических данных они обозначены как "не изолированные": блок питания 24 В DC модуля CPU, силовой вход для катушки реле сигнального модуля или блок питания не изолированного аналогового

входа. Все не изолированные клеммы M должны быть подсоединены к одному и тому же внешнему опорному потенциалу.

⚠ ПРЕДУПРЕЖДЕНИЕ

Подключение не изолированных клемм M к различным опорным потенциалам вызовет появление непредусмотренных токов, которые могут привести к повреждению или к непредсказуемому поведению целевой системы и подключенного оборудования.

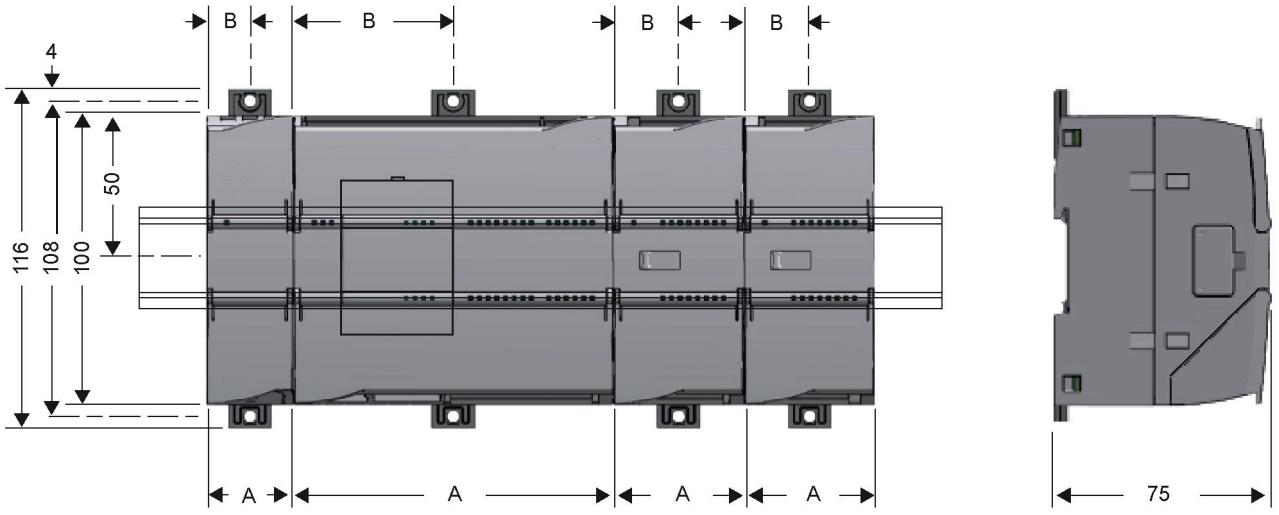
Несоблюдение этих рекомендаций может причинить вред или вызвать непредсказуемое поведение и, как следствие, может привести к тяжким телесным повреждениям и/или материальному ущербу.

Все не изолированные клеммы M в системе S7-1200 всегда должны подключаться к одному и тому же опорному потенциалу.

4.3 Процесс монтажа и демонтажа

4.3.1 Монтажные размеры для устройств S7-1200

CPU 1211C, CPU 1212C, CPU 1214C
(measurements in mm)



CPU 1215C, CPU 1217C
(measurements in mm)

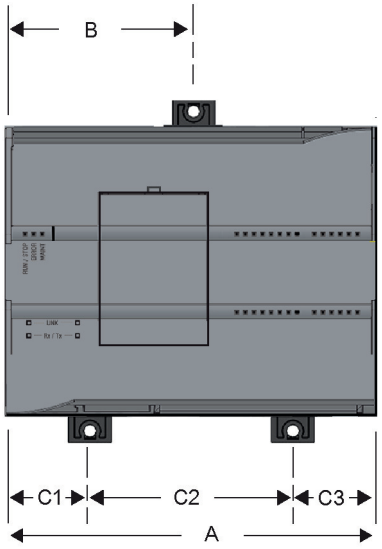


Таблица 4-1 Размеры для монтажа (мм)

Устройства S71200		**Ширина A (мм)**	**Ширина B (мм)**	**Ширина C (мм)**
CPU	CPU 1211C и CPU 1212C	90	45	--
	CPU 1214C	110	55	--
	CPU 1215C	130	65 (верх)	Низ: C1: 32,5 C2: 65 C3: 32,5
	CPU 1217C	150	75	Низ: C1: 37,5 C2: 75 C3: 37,5
Сигнальные модули	8 или 16 цифровых I/O 2, 4 или 8 аналоговых I/O Термопара, 4 или 8 I/O RTD, 4 I/O SM 1278 IO Link-Master	45	22,5	--
	8 цифровых выходов x реле (переключение)	70	35	--
	16 аналоговых I/O RTD, 8 I/O	70	35	--
	SM 1238 Energy Meter	45	22,5	--
Коммуникационные интерфейсы	CM 1241 RS232 и CM 1241 RS422/485 CM 1243-5 PROFIBUS-Master и CM 1242-5 PROFIBUS-Slave CM 1242-2 AS-i-Master CP 1242-7 GPRS V2 CP 1243-7 LTE-US CP 1243-7 LTE-EU CP 1243-1 CP 1243-8 IRC RF120C	30	15	--
	TS (TeleService) адаптер IE Advanced [1] TS (TeleService) адаптер IE Basic [1] TS адаптер TS модуль	 30 30	 15 15	 -- --

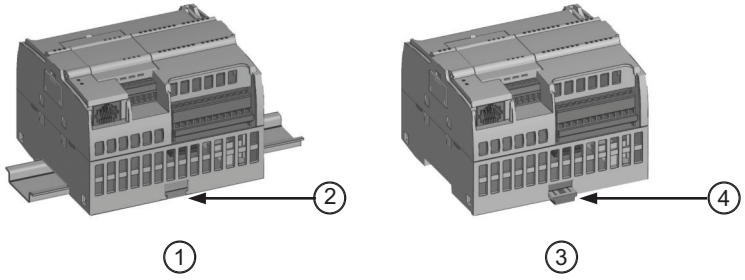
[1] Прежде, чем установить TS (TeleService) адаптер IE Advanced или IE Basic, необходимо подсоединить TS адаптер и TS модуль. Полная ширина ("ширина A") составляет 60 мм.

Все CPU, SM, CM и CP могут монтироваться на DIN-рейку или на пластину. Для крепления устройства на DIN-рейке используйте шинные зажимы. Эти зажимы защелкиваются и в извлеченном положении, чтобы сделать возможным монтаж устройства на пластину. Внутренние размеры отверстия для шинных зажимов на устройстве составляет 4,3 мм.

Над и под устройством в качестве защиты от перегрева необходимо оставить свободное пространство в 25 мм для свободной циркуляции воздуха.

Монтаж и демонтаж устройств S71200

Модуль CPU легко устанавливается на стандартной DIN-рейке в пульте управления. Для крепления устройства на DIN-рейке с ним поставляются шинные зажимы. Эти зажимы могут быть защелкнуты также и в извлеченном положении, делая возможным крепление устройства винтами в пульте управления.



① Монтаж на DIN-рейку
② Стандартный шинный зажим в запертом положении
③ Монтаж на пластину
④ Зажим в извлеченном положении для монтажа на пластину

Перед монтажом или демонтажем электрического устройства необходимо убедиться, что оно обесточено. Кроме этого проследить, чтобы были выключены и все подсоединенные устройства

⚠ ПРЕДУПРЕЖДЕНИЕ

Монтаж или демонтаж S7-1200 или подсоединенных к нему устройств во включенном состоянии может

вызвать поражение электрическим током или непредусмотренное поведение оборудования.

Следует предпринять все необходимые меры безопасности и перед монтажом или демонтажем устройства убедиться, что питание CPU S7-1200 выключено.

При монтаже/демонтаже устройства S7-1200 всегда следует проверить, используется ли правильный модуль / устройство.

⚠ ПРЕДУПРЕЖДЕНИЕ

Неправильная установка модуля S7-1200 может привести к непредсказуемому поведению программы S7-1200.

Ошибка при выборе модификации устройства S7-1200 для замены, при его ориентации или нарушение порядка монтажа, может стать причиной летального исхода или тяжких травм персонала и/или материального ущерба из-за непредсказуемого поведения устройства.

Для замены всегда следует использовать устройство S7-1200 той же модификации, смонтировать его в правильном месте и точно ориентировать.

⚠ ПРЕДУПРЕЖДЕНИЕ

Не отключать оборудование при наличии легковоспламеняющейся или горючей атмосферы.

Отключение оборудования при наличии легковоспламеняющейся или горючей атмосферы может привести к пожару или взрыву и, как следствие, к смерти, серьезным травмам и / или материальному ущербу.

Соблюдение соразмерных меры безопасности является необходимым условием при наличии легковоспламеняющейся или горючей атмосферы.

Примечание

Электростатические разряды могут повредить устройство или предусмотренный для него слот в CPU.

При работе с устройством следует стоять на электропроводящей, заземленной поверхности и/или носить заземленный браслет.

4.3.2 Монтаж и демонтаж CPU

Модуль CPU может быть установлен на пластину или на стандартную DIN–рейку.

Примечание

Подсоединить коммуникационные модули к CPU и смонтировать всю сборку как один узел. Установить сигнальные модули только после монтажа CPU.

При монтаже устройств на DIN-рейку или на пластину необходимо обратить внимание на следующее:

- При установке на DIN-рейку проследить, чтобы верхний шинный зажим CPU и подсоединенных CM находился в защелкнутом (задвинутом) положении, а нижний шинный зажим - в выдвинутом положении.
- После монтажа на DIN-рейку защелкнуть (задвинуть) нижние шинные зажимы для фиксации устройств на DIN-рейке.
- При монтаже на пластину необходимо помнить, что шинные зажимы должны находиться в выдвинутом положении.

Для установки CPU на пластину действовать следующим образом:

1. Изготовить отверстия (M4) с параметрами, перечисленными в таблице Размеры для монтажа (мм) (Страница 57).
2. Убедиться, что CPU и все устройства S7-1200 отсоединены от электропитания.
3. Извлечь монтажные зажимы из модуля. Необходимо помнить, что шинные зажимы сверху и снизу на CPU должны находиться в выдвинутом положении.
4. Закрепить модуль с помощью винта со шлицевой головкой M4, используя пружинное кольцо и плоскую подкладную шайбу. Не использовать винты с потайной головкой.

Примечание

Тип винта зависит от материала, на который монтируется модуль. Затянуть винт с соответствующим моментом, чтобы пружинное кольцо приняло плоскую форму. Не перетягивать винты. Не использовать винты с потайной головкой.

Примечание

В случае сильных вибраций в месте установки или при вертикальном расположении CPU рекомендуется использовать DIN-шинные скобы для фиксации CPU на рейке. Установить концевой фиксатор (8WA1808 или 8WA1805) на DIN-рейку, чтобы прижать модули друг к другу. При сильных вибрациях рекомендуется устанавливать CPU на пластину для лучшей защиты.

Таблица 4-2 Монтаж CPU на DIN-рейку

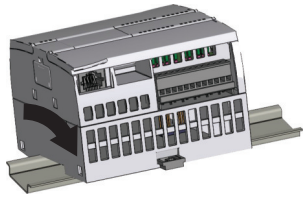
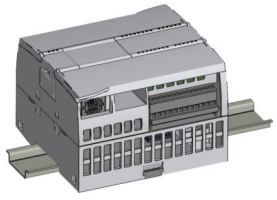
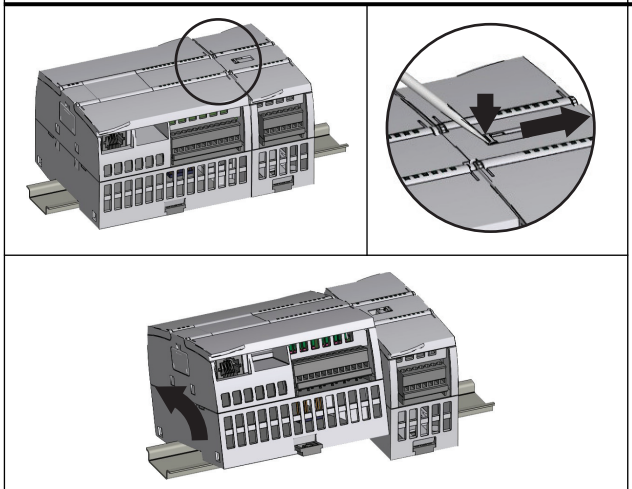
Постановка задачи	Порядок действий
 	1. Смонтировать DIN-рейку. Прикрутить DIN-рейку к монтажной панели через каждые 75 мм. 2. Убедиться, что CPU и все устройства S7-1200 отсоединены от электропитания. 3. Навесить CPU сверху на DIN-рейку. 4. Извлечь шинный зажим в нижней части CPU, чтобы CPU мог плотно прилечь к рейке. 5. Повернуть CPU вниз в монтажное положение на рейке. 6. Вдавите зажимы, чтобы закрепить CPU на рейке.

Таблица 4-3 Демонтаж CPU с DIN-рейки

Постановка задачи	Порядок действий
	1. Убедиться, что CPU и все устройства S7-1200 отсоединены от электропитания. 2. Отсоединить разъемы I/O, провода и кабели от CPU (Страница 67). 3. Снять CPU и прикрепленные к нему коммуникационные модули как один узел. Все сигнальные модули должны остаться на рейке. 4. Если к CPU подключен SM, то отвести шинный соединитель: – Установить отвертку рядом с ползунком на верхней стороне сигнального модуля. – Нажать вниз, чтобы разъединить CPU и соединитель. – Передвинуть ползунок вправо до упора. 5. Снять CPU: – Извлечь шинный зажим в нижней части CPU, чтобы освободить CPU от рейки. – Повернуть CPU вверх и в сторону от шины и снять его.

4.3.3 Монтаж и демонтаж SB, CB или BB

Таблица 4-4 Монтаж SB, CB или BB 1297

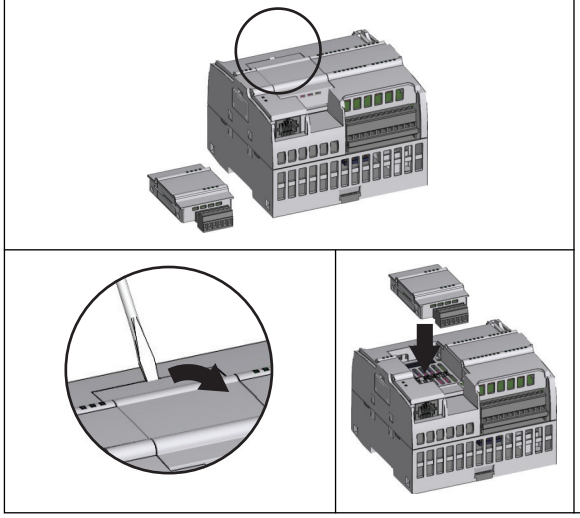
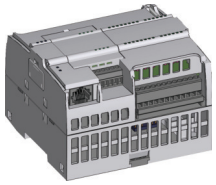
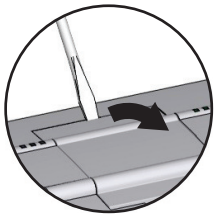
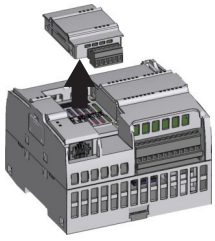
Постановка задачи	Порядок действий
	1. Убедиться, что CPU и все устройства S7-1200 отсоединены от электропитания. 2. Снять верхнюю и нижнюю крышку терминального блока с CPU. 3. Вставить отвертку в щель сверху на CPU с задней стороны крышки. 4. Осторожно приподнять крышку вверх, чтобы освободить ее из держателя, и снять ее с CPU. 5. Вставить модуль сверху в его монтажное положение в CPU. 6. Нажать на модуль сверху до щелчка для фиксации. 7. Снова установить крышки терминального блока.

Таблица 4-5 Демонтаж SB, CB или BB 1297

Постановка задачи	Порядок действий
  	1. Убедиться, что CPU и все устройства S7-1200 отсоединены от электропитания. 2. Снять верхнюю и нижнюю крышку терминального блока с CPU. 3. Отсоединить разъем сигнальной платы (при наличии), аккуратно отодвинув его отверткой. 4. Вставить отвертку в щель на верхней стороне модуля. 5. Осторожно освободить модуль из держателя и отсоединить его от CPU. 6. Извлечь модуль не используя при этом отвертку вертикально вверх из монтажного положения и из CPU. 7. Снова установить крышку CPU. 8. Снова установить крышки терминального блока.

Установка или замена батареи в плате буферной батареи BB 1297

В BB 1297 используется батарея типа CR1025. Батарея не входит в комплект поставки BB 1297 и должна приобретаться отдельно. Для установки или замены батареи следует выполнить следующие шаги:

1. Установить новую батарею BB 1297 так, чтобы положительная сторона батареи находилась сверху, а отрицательная - рядом с печатной монтажной платой.
2. Теперь BB 1297 может быть установлена в CPU. Убедиться, что CPU и все устройства S7-1200 отсоединены от электропитания и установить BB 1297 согласно описанию выше.

Для замены батареи BB 1297 действовать следующим образом:

1. Убедиться, что CPU и все устройства S7-1200 отсоединены от электропитания. Демонтировать BB 1297 из CPU согласно описанию выше.
2. Осторожно извлечь старую батарею с помощью маленькой отвертки. Освободить батарею из зажима.
3. Установить новую батарею CR1025 так, чтобы положительная сторона батареи находилась сверху, а отрицательная - рядом с печатной монтажной платой.
4. Снова смонтировать плату буферной батареи BB 1297 согласно описанию выше.

⚠ ПРЕДУПРЕЖДЕНИЕ

Установка не специфицированной батареи в плату BB 1297, или подключение не специфицированной батареи к цепи любым другим способом могут привести к возгоранию или повреждению компонентов, а также непредсказуемой работе оборудования.

Возгорание или непредсказуемая работа оборудования могут стать причиной серьезных травм или материального ущерба.

Для буферизации часов реального времени следует использовать только специфицированную батарею CR1025.

4.3.4 Монтаж и демонтаж SM

Таблица 4-6 Монтаж SM

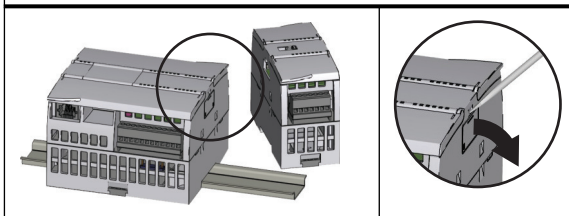
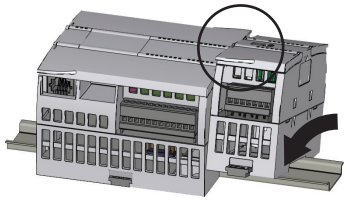
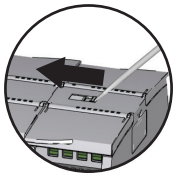
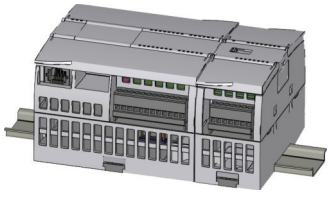
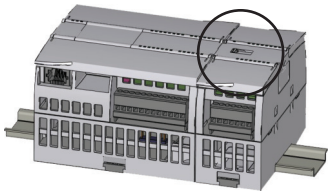
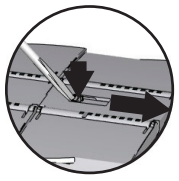
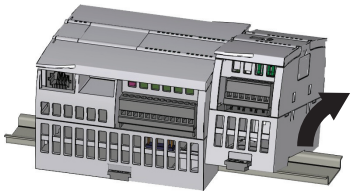
Постановка задачи	Порядок действий
	SM монтируется после монтажа CPU. 1. Убедиться, что CPU и все устройства S7-1200 отсоединены от электропитания. 2. Снять крышку для подключения с правой стороны модуля CPU: – Вставить отвертку в щель над крышкой. – Осторожно освободить крышку из держателя вверху, и снять ее. 3. Отложить крышку для последующего использования.
 	Подключение SM к CPU: 1. Поместить SM рядом с CPU. 2. Навесить CPU сверху на DIN-рейку. 3. Извлечь нижний шинный зажим, чтобы SM мог плотно прилечь к рейке. 4. Повернуть SM вниз, чтобы он оказался рядом с CPU, и задвинуть нижний зажим, чтобы SM прочно сидел на рейке.
	Шинный соединитель отвечает за механическое и электрическое подсоединение SM. 1. Установить отвертку рядом с ползунком на верхней стороне SM. 2. Передвинуть ползунок влево до упора, чтобы вставить шинный соединитель в CPU. Повторить процесс для пристраивания одно сигнального модуля к другому.

Таблица 4-7 Демонтаж SM

Постановка задачи	Порядок действий
	Можно демонтировать любой SM, не демонтируя при этом CPU или другие SM. 1. Убедиться, что CPU и все устройства S7-1200 отсоединены от электропитания. 2. Отсоединить разъем I/O и проводку от SM (Страница 67). 3. Отвести шинный соединитель – Установить отвертку рядом с ползунком на верхней стороне SM. – Нажать вниз, чтобы разъединить CPU и соединитель. – Передвинуть ползунок вправо до упора. Если справа установлен еще один SM, то повтрить процесс для этого SM.
	
	Снять SM: 1. Выдвинуть нижний шинный зажим, чтобы освободить SM от рейки. 2. Откинуть SM вверх и снять его с рейки. Удалить SM из системы. 3. При необходимости установить крышку на шинный соединитель CPU, чтобы исключить его загрязнение. Повторить процесс для отсоединения одного сигнального модуля от другого.

4.3.5 Монтаж и демонтаж CM или CP

Подсоединить коммуникационные модули к CPU и смонтировать всю сборку как один узел. См. Монтаж и демонтаж CPU (Страница 60).

Таблица 4-8 Монтаж CM или CP

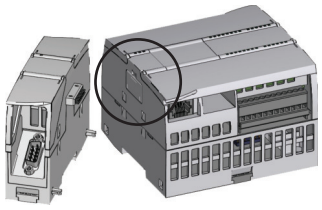
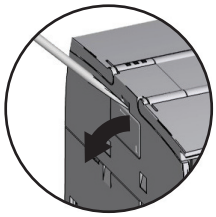
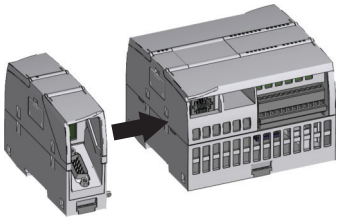
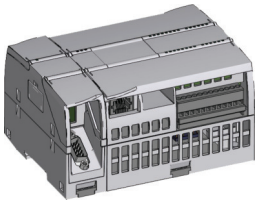
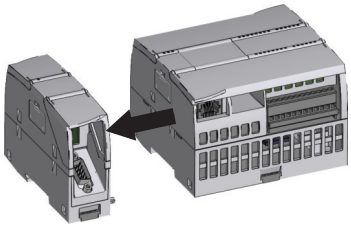
Постановка задачи	Порядок действий
  	1. Убедиться, что CPU и все устройства S7-1200 отсоединены от электропитания. 2. Подсоединить коммуникационные модули к CPU и смонтировать всю сборку как один узел на DIN-рейку или на пластину. 3. Снять крышку шины на левой стороне модуля CPU: – Вставить отвертку в щель над крышкой шины. – Осторожно освободить крышку из держателя вверху. 4. Снять крышку шины. Отложить крышку для последующего использования. 5. Подключение CM или CP к CPU: – Совместить шинный соединитель и штифты CM с отверстиями в CPU. – Крепко прижать блоки друг к другу до защелкивания штифтов. 6. Установить CPU и CP на DIN-рейку или на пластину.

Таблица 4-9 Демонтаж CM или CP

Постановка задачи	Порядок действий
 	Демонтировать CPU и CM как один узел с DIN-рейки или пластины. 1. Убедиться, что CPU и все устройства S7-1200 отсоединены от электропитания. 2. Отсоединить разъемы I/O, провода и кабели от CPU и CM. 3. При монтаже на DIN-рейке выдвинуть нижние шинные зажимы на CPU и CM. 4. Демонтировать CPU и CM как один узел с DIN-рейки или пластины. 5. Крепко удерживая CPU и CM, разъединить их.

ВНИМАНИЕ
Не использовать инструменты для разделения модулей. Не использовать инструмент для разделения модулей, так как это может их повредить.

4.3.6 Демонтаж и монтаж терминального блока S7-1200

Модули CPU, SB и SM снабжены съемными разъемами, облегчающими подключение.

Таблица 4-10 Демонтаж терминального блока

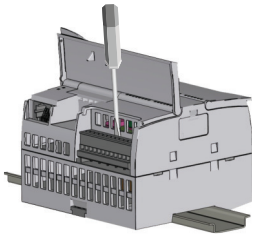
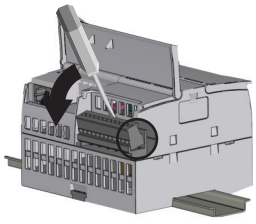
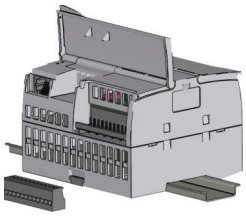
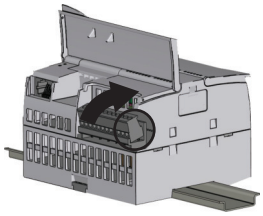
Постановка задачи	**Порядок действий**
 	Для подготовки системы к демонтажу терминального блока следует отключить напряжение питания CPU и открыть крышку над терминальным блоком. 1. Убедиться, что CPU и все устройства S7-1200 отсоединены от электропитания. 2. Найти на верхней стороне терминального блока паз для острия отвертки. 3. Вставить отвертку в паз. 4. Используя отвертку в качестве рычага, осторожно освободить верхнюю часть терминального блока из фиксатора в CPU. При освобождении терминального блока будет слышен щелчок. 5. Извлечь терминальный блок из CPU вручную.

Таблица 4-11 Монтаж терминального блока

Постановка задачи	Порядок действий
 	Подготовить компоненты для монтажа терминального блока, отключив питание CPU и открыв крышку терминального блока. 1. Убедиться, что CPU и все устройства S7-1200 отсоединены от электропитания. 2. Совместить блок с контактами на устройстве. 3. Точно установить сторону подключения терминального блока внутри штекерной колодки. 4. Нажать на терминальный блок, одновременно разворачивая его вниз, до щелчка. Проверить, правильно ли расположен терминальный блок и надежно ли он закреплен.

4.3.7 Монтаж и демонтаж расширительного кабеля

Расширительный кабель S7-1200 обеспечивает дополнительную гибкость при конфигурировании системы S7-1200. В системе CPU можно использовать только один расширительный кабель. Расширительный кабель устанавливается между CPU и первым SM или между двумя SM.

Таблица 4-12 Монтаж и демонтаж штекерной части расширительного кабеля

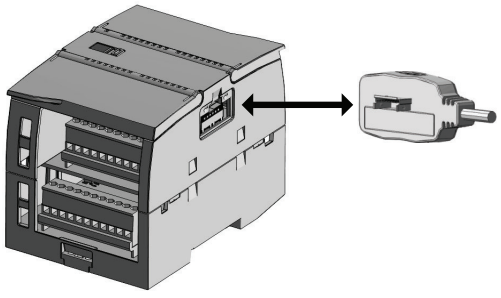
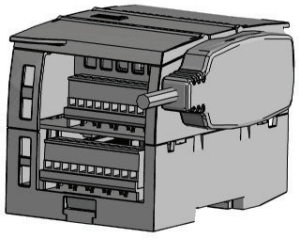
Постановка задачи	Порядок действий
 	Монтаж штекерной части: 1. Убедиться, что CPU и все устройства S7-1200 отсоединены от электропитания. 2. Вставить штекерную часть в шинный разъем на правой стороне сигнального модуля или CPU. Демонтаж штекерной части: 1. Убедиться, что CPU и все устройства S7-1200 отсоединены от электропитания. 2. Извлечь штекерную часть из сигнального модуля или из CPU.

Таблица 4-13 Монтаж гнездовой части расширительного кабеля

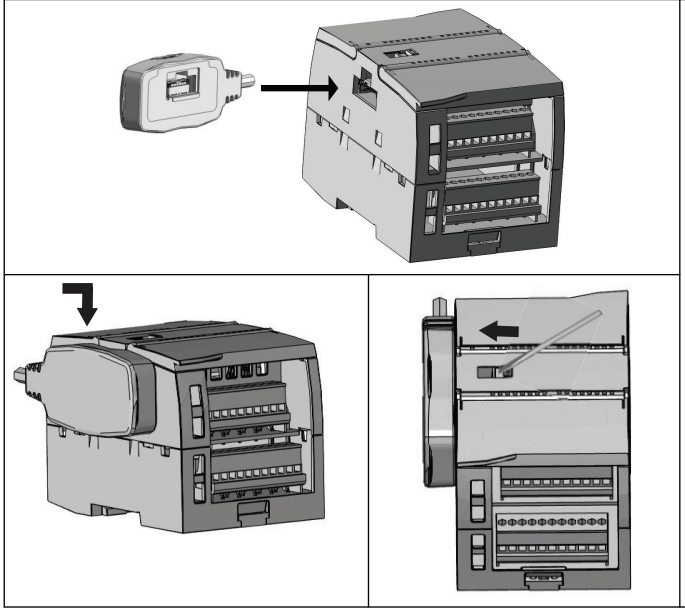
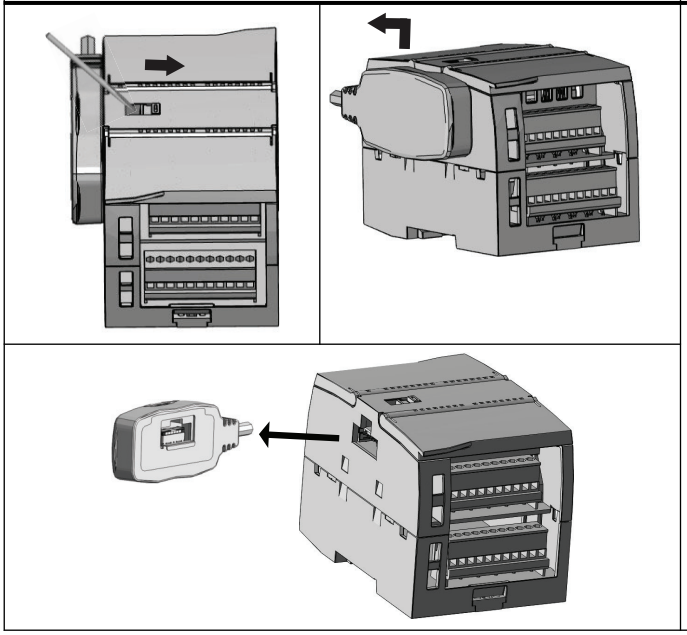
Постановка задачи	**Порядок действий**
	1. Убедиться, что CPU и все устройства S7-1200 отсоединены от электропитания. 2. Вставить гнездовую часть в шинный разъем на левой стороне сигнального модуля. 3. Вставить выступ гнездовой части с крючком в корпус на шинном разъеме и осторожно нажать для защелкивания крючка. 4. Закрепить штекерное соединение: – Установить отвертку рядом с ползунком на верхней стороне сигнального модуля. – Передвинуть ползунок влево до упора. Для фиксации штекерного соединения необходимо передвинуть ползунок влево до упора. Ползунок-фиксатор должен защелкнуться.

Таблица 4-14 Демонтаж гнездовой части расширительного кабеля

Постановка задачи	**Порядок действий**
	1. Убедиться, что CPU и все устройства S7-1200 отсоединены от электропитания. 2. Разъединение разъема: – Установить отвертку рядом с ползунком на верхней стороне сигнального модуля. – Слегка нажав на отвертку, передвинуть ползунок вправо до упора. 3. Немного приподнять соединитель, чтобы освободить крючок. 4. Снять гнездовую часть.

Примечание

Монтаж расширительного кабеля при воздействии на модуль вибрации

Если расширительный кабель подключается к модулями, которые перемещаются или не закреплены, то стопорное соединение на штекерной части кабеля со временем может ослабнуть.

Для дополнительной компенсации натяжения следует зафиксировать конец кабеля со штекерной частью с помощью кабельной стяжки на DIN-рейке (или на другой опоре).

Избегать чрезмерного применения силы при протяжке кабеля в процессе монтажа. После завершения монтажа необходимо убедиться, что соединение между кабелем и модулем находится в правильном положении.

4.3.8 TS (TeleService) адаптер

4.3.8.1 Подключение TS адаптера

Перед монтажом TS адаптера IE Advanced или IE Basic, необходимо подсоединить TS адаптер и TS модуль.

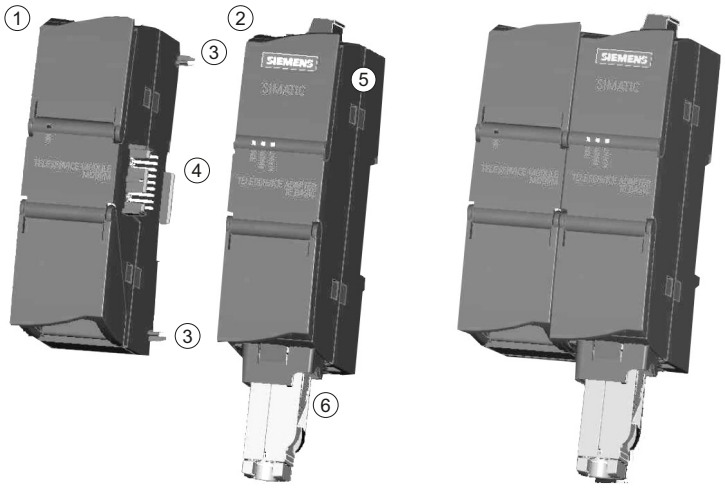
Доступные TS модули:

- TS модуль RS232
- TS модуль модема
- TS модуль GSM
- TS модуль ISDN

Примечание

Прикосновение к контактам разъема ④ TS модуля может вызвать повреждения модуля.

Придерживаться мер по защите элементов, подверженных опасности разрушения в результате электростатического заряда, чтобы исключить повреждение TS модуля электростатическим разрядом. Перед подсоединением TS модуля и TS адаптера следует убедиться, что они находятся в нерабочем состоянии.



①	TS модуль	④	Разъем TS модуля
②	TS адаптер	⑤	Не может быть открыт
③	Элементы	⑥	Ethernet порт

Примечание

Перед подключением TS-модуля и TS адаптера следует убедиться, что контактные штырьки ④ не погнуты.

При выполнении соединения следует убедиться, что штекерная часть и направляющие штифты расположены правильно.

Подключить TS модуль к TS адаптеру. Ни использовать силу при подключении TS адаптера к другому устройству, напр., S7-1200 CPU. Не вносить изменений в механическую конструкцию разъема и не удалять / не повреждать направляющие штифты.

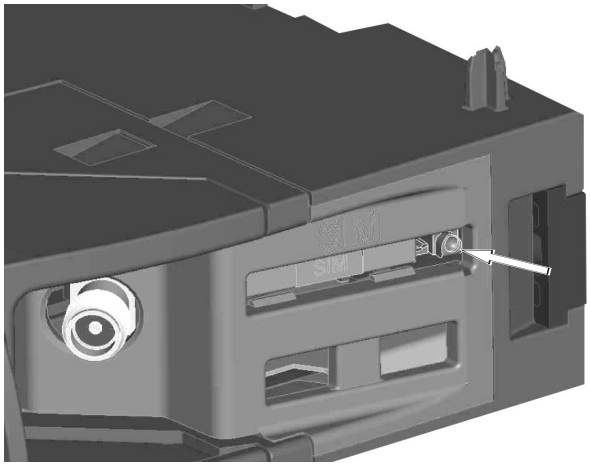
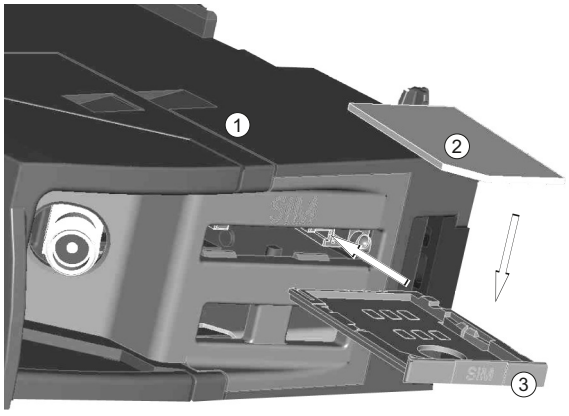
4.3.8.2 Установка SIM карты

Найти слот для SIM карты на нижней стороне TS module GSM.

Примечание

Можно извлекать или вставлять SIM карту только при отключенном TS module GSM.

Таблица 4-15 Установка SIM карты

Постановка задачи	Порядок действий
	Нажать острым предметом на кнопку извлечения лотка SIM карты (в направлении стрелки) и вынуть лоток SIM карты.
	Поместить SIM карту в лоток, как показано на рисунке, и снова вставить лоток SIM-карты в слот. ①TS Module GSM ② SIM карта ③ Лоток SIM карты

Примечание

Убедиться, что SIM карта правильно расположена в лотке. В противном случае, SIM карта не соединится с модулем, и кнопка выброса может не извлекать лоток карты.

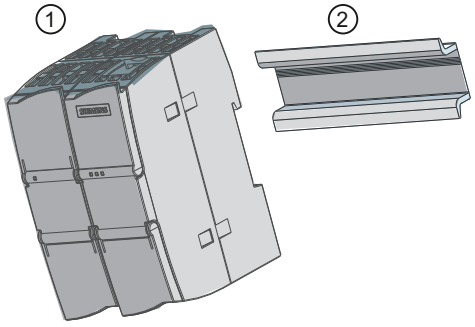
4.3.8.3 Монтаж TS адаптера на DIN-рейку

Необходимые условия: TS адаптер и TS модуль должны быть соединены вместе, и DIN-рейка должна быть установлена.

Примечание

При установке устройства дистанционного обслуживания (TS узла) вертикально или в среде с повышенной вибрацией может случиться, что TS модуль отсоединится от TS адаптера. Установить концевой фиксатор 8WA1808 на DIN-рейку, чтобы прижать модули друг к другу.

Таблица 4-16 Монтаж и демонтаж TS адаптера

Постановка задачи	Порядок действий
	Монтаж: 1. Навесить TS адаптер с подключенным TS модулем ① на DIN-рейку ②. 2. Защелкнуть узел на рейке, повернув его вниз. 3. Передвинуть внутрь зажимы для DIN рейки на каждом из модулей, чтобы закрепить их на рейке.
	Демонтаж: 1. Отключить аналоговый кабель и Ethernet кабель в нижней части TS адаптера. 2. Выключить питание TS адаптера. 3. С помощью отвертки освободить зажимы для рейки на обоих модулях. 4. Повернуть узел вверх, чтобы снять его с DIN-рейки.

⚠ ПРЕДУПРЕЖДЕНИЕ

Правила безопасности при монтаже и демонтаже TS адаптера.

Перед отключением питания узла, следует снять заземление TS адаптера путем отсоединения аналогового кабеля и Ethernet-кабеля. Отказ от соблюдения этой предосторожности может стать причиной летального исхода или тяжких травм персонала и/или материального ущерба из-за непредсказуемого поведения устройства.

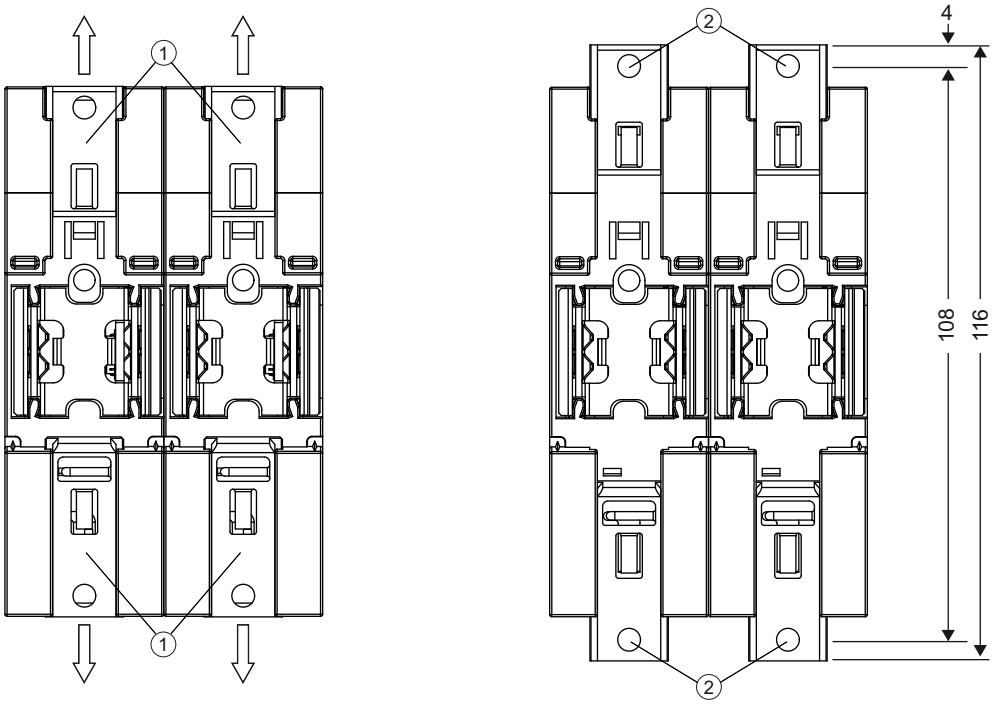
Это правило безопасности является обязательным к соблюдению при каждом монтаже иди демонтаже TS адаптера.

4.3.8.4 Монтаж TS адаптера на пластину

Необходимые условия: TS адаптер и TS модуль должны быть соединены вместе.

1. Сдвинуть крепежный ползунок ① в сторону от TS адаптера и TS модуля в направлении по стрелке до фиксации.
2. Закрепить TS адаптер и TS модуль в обозначенных ② позициях винтами на предназначенной для монтажа стене.

Рисунок ниже показывает TS адаптер сзади, с ползунками ① в обеих позициях:



① Крепежный ползунок
② Отверстия для монтажа на стену

4.4 Правила электромонтажа

Надлежащее заземление и подключение всего электрического оборудования играет важную роль в обеспечении оптимального функционирования системы и увеличении стойкости к воздействию помех для приложения и S7-1200. Схемы соединений для S7-1200 можно найти в Технических данных (Страница 1433).

Необходимое условие

Перед заземлением или электромонтажом электрического устройства необходимо убедиться, что оно обесточено. Кроме этого проследить, чтобы были выключены и все подсоединенные устройства

При подключении S7-1200 и всех присоединенных к нему устройств следует обеспечить выполнение всех действующих и обязательных стандартов/норм. При монтаже и эксплуатации оборудования следует придерживаться действующих национальных и

региональных правил. Выяснить у местных властей, какие стандарты/нормы и регламенты действуют в каждом конкретном случае.

⚠ ПРЕДУПРЕЖДЕНИЕ

Монтаж или подключение S7-1200 или подсоединенных к нему устройств во включенном состоянии может вызвать поражение электрическим током или непредусмотренное поведение оборудования.

Отказ отключить питание S7-1200 и относящегося к нему оборудования в процессе монтажа или демонтажа может стать причиной летального исхода, тяжких травм персонала и/или материального ущерба из-за поражения электрическим током или непредсказуемого поведения устройства.

Следует предпринять все необходимые меры безопасности перед монтажом или демонтажем устройства и убедиться, что питание S7-1200 выключено.

При планировании заземления и подключения системы S7-1200 всегда следует помнить о безопасности. Электронные устройства управления, например, S7-1200, могут выходить из строя и вызвать непредсказуемое поведение контролируемого и управляемого оборудования. Поэтому следует использовать предохранительные устройства, не зависимые от S7-1200, чтобы избежать возможных травм персонала и/или материального ущерба.

⚠ ПРЕДУПРЕЖДЕНИЕ

Устройства управления могут выходить из строя в небезопасных рабочих состояниях, вызывая тем самым непредсказуемое поведение управляемого оборудования.

Такое непредсказуемое поведение может стать причиной летального исхода, тяжких травм персонала и/или материального ущерба.

Поэтому следует позаботиться о функции АВАРИЙНОГО ОТКЛЮЧЕНИЯ, электромеханических или других резервируемых устройствах обеспечения безопасности, не зависящих от S7-1200.

Руководство по изоляции

Граничные значения для источника питания переменного тока и для входов/выходов цепей переменного тока S7-1200 рассчитаны и допущены для обеспечения надежной электрической развязки (изоляции) между напряжениями сети переменного тока и низковольтными цепями. В зависимости от стандарта/нормы, эти граничные значения требуют двойной или усиленной изоляции или основной плюс дополнительной изоляции. Компоненты, находящиеся вне этих границ, например, оптопары, конденсаторы, трансформаторы и реле, допущены к эксплуатации как устройства, обеспечивающие надежную электрическую развязку. Только электрические цепи, классифицированные для сетевого напряжения переменного тока, включают в себя надежную изоляцию по отношению к другим цепям тока. Границы развязки между цепями 24 В постоянного тока являются исключительно функциональными и не должны рассматриваться как критерии безопасности.

Выход питания датчиков, коммуникационные цепи и электрические цепи внутренней логики S7-1200 со встроенным источником питания переменного тока в соответствии с EN 61131-2 получают питание как цепи безопасного сверхнизкого напряжения (SELV, safety extra-low voltage).

Для поддержания безопасного характера низковольтных цепей S7-1200 внешние соединения с коммуникационными портами, аналоговыми цепями и всеми источниками питания с номинальным напряжением 24 В DC, а также с цепями ввода / вывода, должны получать питание от сертифицированных согласно различным стандартами/нормам источников, удовлетворяющих требованиям SELV, PELV, класс 2, с ограничением по напряжению и по мощности.

⚠ ПРЕДУПРЕЖДЕНИЕ

Использование не изолированных или имеющих одиночную изоляцию источников для питания низковольтных цепей из линий переменного тока, может привести к появлению опасных напряжений в контурах, которые считаются безопасными для прикосновения, например, в линиях связи и в проводке низковольтных датчиков.

Такие неожиданно высокие напряжения могут вызвать поражение электрическим током и стать причиной летального исхода, тяжких травм и/или материального ущерба.

Следует использовать только такие преобразователи высокого напряжения в низкое, которые сертифицированы как источники для безопасных при прикосновении цепей с ограниченным напряжением.

Правила заземления для S7-1200

Наилучшим методом для заземления приложения является подключение всех общих соединений и всех клемм заземления системы S7-1200 и всех подключенных к ней устройств к одной точке. Эта точка должна быть напрямую соединена с системной землей.

Провода заземления должны быть по возможности короткими и иметь жилы с большим поперечным сечением, например. 2 мм^2 (14 AWG).

При выборе точек заземления следует принять во внимание соответствующие предписания по технике безопасности и обеспечить надлежащее функционирование защитных отключающих устройств.

Правила электромонтажа для S7-1200

При проектировании системы электрических соединений для S7-1200 следует предусмотреть единый разъединитель, который одновременно снимает напряжение с блока питания CPU S7-1200 и всех входных / выходных цепей. Необходимо наличие максимальной токовой защиты, например, предохранителя или автоматический выключатель, чтобы ограничить аварийный ток в питающей проводке. Можно предусмотреть дополнительную защиту с помощью предохранителей или других ограничителей тока в каждой из выходных цепей.

Установить подходящие ограничители перенапряжений для любых кабелей, которые могут пострадать в результате удара молнии. Более подробная информация содержится

в разделе "Общее" технических данных в Устойчивость к скачкам напряжения (Страница 1433).

Избегать прокладки низковольтных сигнальных кабелей и коммуникационных кабелей в одном канале с кабелями питания переменного тока и часто коммутируемыми силовыми кабелями постоянного тока. Всегда прокладывать провода парами: нейтраль или нулевой провод вместе с фазой или сигнальным проводом.

Использовать наиболее короткие из возможных проводов и выбирать сечение, соответствующее току.

Рабочая температура проводов и кабелей должна быть на 30 °C выше температуры окружающей среды для S7-1200 (пример: кабели/провода, которые будут использоваться при температуре окружающей среды в 55 °C, должны быть рассчитана как минимум на 85 °C). Следует определить другие типы проводки и требования к материалу на основе установленных расчетных значений цепей тока и среды инсталляции.

Использовать экранированные провода для оптимальной защиты от электрических помех. Как правило, заземление экрана у S7-1200 дает лучшие результаты. Следует заземлить экран коммуникационных кабелей вместе с корпусами коммуникационных штекеров S7-1200 через разъемы, имеющие токопроводящее соединение с экраном кабеля, или соединяя экран коммуникационного кабеля с отдельной землей. Необходимо заземлить экраны других кабелей , используя зажимы или медную ленту вокруг экрана, чтобы обеспечить соединение с заземляющей точкой по большой площади.

При подключении входной цепи, запитанной от внешнего источника, следует использовать устройство максимальной токовой защиты во входной цепи. Для цепей, которые запитаны от источника 24 В DC для питания датчиков в S7-1200, внешняя защита не нужна, т.к. источник питания датчиков уже имеет схему ограничения тока.

Все модули S7-1200 оснащены съемными терминальными блоками для подключений пользователя. Для предотвращения ослабления контактов следует убедиться, что терминальный блок вставлен плотно, и провода в нем надежно закреплены.

Система S7-1200 работает в границах, определяемых потенциальной развязкой, что препятствует возникновению нежелательных токов в установке. При планировании подключения для системы следует принять во внимание эти границы. В главе Технические данные (Страница 1499) можно найти параметры для доступной потенциальной развязки и её границах. Электрические цепи, классифицированные для сетевого напряжения переменного тока, включают в себя надежную изоляцию по отношению к другим цепям тока. Границы развязки между цепями 24 В постоянного тока являются исключительно функциональными и не должны рассматриваться как критерии безопасности.

Сводная таблица правил подключения для S7-1200 CPU, SM и SB представлена ниже:

Таблица 4-17 Правила подключения для S7-1200-CPU, -SM и -SB

Правила подключения для ...	**Соединитель CPU и SM**		**Соединитель SB**
Соединительная техника	**Push-in клемма**	**Винт**	**Винт**
Подключаемое сечение вывода для стандартных жил	2 мм² до 0,3 мм² (14 AWG до 22 AWG)		1,3 мм² до 0,3 мм² (16 AWG до 22 AWG)
Число жил на соединение	1 или комбинация из 2 жил в кабеле с двойной оболочкой до 2 мм² (общее)		1 или комбинация из 2 жил до 1,3 мм² (общее)
Длина зачистки изоляции	Использование оконечных кабельных муфт для надежного электрического соединения	6,4 мм	От 6,3 до 7 мм
Момент затяжки* (максимальный)	-/-	0,56 Нм	0,33 Нм
Инструмент	Шлицевая отвёртка от 2,5 до 3,0 мм		

* Не перетягивать винты, чтобы не повредить соединение.

Примечание

Использование оконечных кабельных муфт для многожильных проводов уменьшает опасность коротких замыканий из-за необжатых жил. Оконечные кабельные муфты, длина которых превышает рекомендуемую длину зачистки изоляции, должны иметь изолирующий бортик для предотвращения КЗ из-за боковых перемещений провода. Ограничения по площади сечения для неизолированных проводов относятся и к оконечным кабельным муфтам.

См. также

Технические данные (Страница 1433)

Правила для ламповых нагрузок

Ламповые нагрузки, включая светодиодные лампы, повреждают контакты реле из-за сильного броска тока при включении. Такой бросок тока обычно в 10 – 15 выше стационарного тока лампы накаливания. Для часто включаемых в течение срока службы приложения ламповых нагрузок рекомендуется использовать сменное промежуточное реле или ограничитель бросков тока.

Правила для индуктивных нагрузок

Блоки схемной защиты должны использовать в случае индуктивных нагрузок для ограничения повышения напряжения при отключении управляющего выхода. Блоки схемной защиты защищают выходы от преждевременного выхода из строя, вызванного переходным процессом высокого напряжения, который происходит, когда электрический ток прерывается индуктивной нагрузкой.

Кроме этого, блоки схемной защиты ограничивают электрические помехи, возникающие при коммутации индуктивных нагрузок. Высокочастотные помехи от плохо подавленных индуктивных нагрузок могут нарушить работу PLC. Размещение внешнего блока схемной защиты параллельно нагрузке и около нагрузки, является наиболее эффективным способом уменьшения электрических помех.

Выходы DC в S7-1200 имеют внутренние блоки схемной защиты, достаточные для индуктивных нагрузок в большинстве приложений. Так как релейные контакты выходов S7-1200 могут использоваться для коммутации нагрузок как постоянного, так и переменного тока, внутренняя защита не предусмотрена.

Хорошим решением является использование контакторов и других индуктивных нагрузок, для которых изготовитель предусматривает блоки схемной защиты, интегрированные в нагрузочное устройство или предлагаемые как дополнительная опция. Однако блоки схемной защиты некоторых изготовителей могут быть несовместимы с конкретным приложением. Дополнительный блок схемной защиты может потребоваться для оптимального подавления помех и увеличения срока службы контактов.

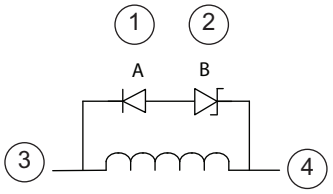
Для нагрузок переменного тока металло-оксидный варистор (MOV) или другой ограничитель напряжения могут использоваться параллельно с RC-цепочкой; такие устройства по отдельности не столь эффективны. Использование MOV без параллельной RC-цепочки часто приводит к возникновению значительных высокочастотных помех с амплитудой вплоть до напряжения стабилизации.

Строго контролируемый переходный процесс при выключении будет иметь частоту "звона" не выше 10 кГц, при этом предпочтительной является частота менее 1 кГц. Пиковое напряжение для линий переменного тока должно быть в пределах +/-1200 В относительно земли. Отрицательное пиковое напряжение для нагрузок постоянного тока с внутренним блоком схемной защиты PLC будет на ~40 В ниже напряжения питания 24 В DC. Внешний блок схемной защиты должен ограничивать переходный процесс до значения в пределах 36 В от источника питания, чтобы разгрузить внутренний блок схемной защиты.

Примечание

Эффективность блока схемной защиты зависит от соответствующего приложения и должна быть проверена в каждом конкретном случае. Следует помнить, что все компоненты должны выбираться под конкретное приложение, для контроля переходного процесса выключение использовать осциллограф.

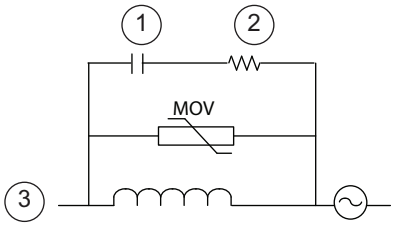
Типовые блоки схемной защиты для DC- или релейных выходов с коммутируемой индуктивной DC нагрузкой



① Диод 1N4001 или эквивалентный
② Стабилитрон 8,2 В (выходы постоянного тока), стабилитрон 36 В (релейные выходы)
③ Выход
④ M, 24 В опорный потенциал

В большинстве случаев достаточно добавить диод (A) через индуктивную нагрузку постоянного тока, но если приложение требует более короткого времени выключения, то рекомендуется использовать стабилитрон (B). Стабилитрон выбирается в соответствии с током в выходной цепи.

Типовые блоки схемной защиты для релейных выходов с коммутируемой индуктивной AC нагрузкой



① Значение для C см. таблицу
② Значение для R см. таблицу
③ Выход

Следует помнить, чтобы рабочее напряжение металлооксидного варистора (MOV) должно быть минимум на 20% выше, чем номинальное напряжение.

Использовать импульсные, неиндуктивные резисторы и конденсаторы, рекомендуемые для импульсных приложений (обычно это металлопленочные). Убедиться, что компоненты отвечают требованиям по средней мощности, пиковой мощности и пиковому напряжению.

Для проектирования своего собственного блока схемной защиты, в следующей таблице можно найти значения резистора и конденсатора для множества AC нагрузок. Эти значения основываются на расчетах с параметрами идеальных компонентов. Значение I эфф. таблице относится к установившемуся току нагрузки в состоянии полного включения.

Таблица 4-18 Значения резисторов и конденсаторов для блоков схемной защиты переменного тока

Индуктивная нагрузка			**Значения для блока схемной защиты**		
I эфф.	230 В AC	120 В AC	Резистор		Конденсатор
A	ВА	ВА	Ω	Вт (ном. мощность)	нФ
0,02	4,6	2,4	15000	0,1	15
0,05	11,5	6	5600	0,25	470
0,1	23	12	2700	0,5	100

Индуктивная нагрузка			Значения для блока схемной защиты		
0,2	46	24	1500	1	150
0,5	115	60	560	2,5	470
1	230	120	270	5	1000
2	460	240	150	10	1500

Значения в таблице отвечают следующим критериям:
Максимальный шаг перехода при выключении < 500 В
Пиковое напряжение резистора < 500 В
Пиковое напряжение конденсатора < 1250 В
Ток блока схемной защиты < 8 % от тока нагрузки (50 Гц)
Ток блока схемной защиты < 11 % от тока нагрузки (60 Гц)
Конденсатор dV/dt < 2 В/мкс
Импульсное рассеяние конденсатора: $\int(dv/dt)^2\ dt < 10000\ В^2/мкс$
Резонансная частота < 300 Гц
Мощность резистора для макс. частоты переключения 2 Гц
Коэффициент мощности 0,3 для типовой индуктивной нагрузки

Правила для дифференциальных входов и выходов

Дифференциальные входы и выходы ведут себя не так, как стандартные входы и выходы. У каждого дифференциального входа и выхода есть два штырьковых вывода. Для определения, включен или выключен дифференциальный вход / выход, необходимо измерить разность потенциалов между этими двумя штырьковыми выводами.

См. подробные технические данные для CPU 1217C в приложении A (Страница 1499).

Основы PLC

5

5.1 Выполнение программы пользователя

CPU поддерживает следующие типы блоков для создания эффективной структуры программы пользователя:

- Организационные блоки (OB) определяют структуру программы. Для некоторых OB предлагается предопределенное поведение и стартовые события, но можно создавать OB и со своими собственными стартовыми событиями.
- Функции (FC) и функциональные блоки (FB) содержат программный код, соответствующий конкретным задачам или комбинациям параметров. Каждая FC / FB предоставляет ряд входных и выходных параметров для совместного использования данных с вызывающим блоком. Кроме этого FB использует еще один блок данных (DB экземпляра) для хранения значений данных для соответствующего экземпляра вызова FB. Один FB может быть вызван многократно, каждый раз с однозначным DB экземпляра. Вызовы одного и того же FB с различными DB экземпляра не сказывается на значениях данных в любом из других DB экземпляра.
- Блоки данных (DB) хранят данные, которые могут использоваться программными блоками.

Обработка программы пользователя начинается с одного или нескольких опциональных пусковых OB, которые однократно выполняются после перехода в рабочее состояние RUN, затем следует один или несколько OB программного цикла, которые выполняются циклически. Блок OB также может быть связан с прерывающим событием, здесь речь может идти о стандартном событии или ошибке. Эти OB выполняются при наступлении соответствующего стандартного события или ошибки.

Функция (FC) или функциональный блок (FB) – это блок с кодом программы, который может быть вызван из OB или из другой функции или другого функционального блока. Возможна следующая глубина вложенности:

- 16 из OB циклов или пусковых OB
- 6 из любых OB прерывающих событий
  Примечание: Программы обеспечения безопасности используют два уровня вложенности. Таким образом, программа пользователя имеет в программах обеспечения безопасности четыре уровня вложенности.

FC не согласованы с конкретным блоком данных (DB). FB непосредственно связаны с DB и используют его для передачи параметров и сохранения промежуточных значений и результатов.

Размер программы пользователя, данных и конфигурации ограничен имеющейся в распоряжении загрузочной памятью и рабочей памятью в CPU. Количество отдельных OB, FC, FB и DB не ограничено. Но общее число блоков не должно превышать 1024.

Каждый цикл включает в себя запись выходов, чтение входов, выполнение инструкций программы пользователя и фоновую обработку. Этот цикл называется также циклом сканирования или просто сканированием.

Решение для автоматизации на базе S7-1200 может состоять из центральной стойки с CPU S7-1200 и дополнительными модулями. Термин "центральная стойка" относится к сборке из CPU и связанных с ним модулей на рейке или пластине. Модули (SM, SB, BB, CB, CM или CP) обнаруживаются и регистрируются только при включении питания.

- Вставка или удаление модуля в центральной стойке при включенном устройстве невозможно. Никогда не вставлять и не удалять модули из центральной стойки при включенном CPU.

  ⚠ ПРЕДУПРЕЖДЕНИЕ

  Меры безопасности при вставке или удалении модулей

  Вставка или удаление модуля (SM, SB, BB, CD, CM или CP) из центральной стойки при включенном CPU может вызвать неполадки или непредсказуемое поведение, и, как следствие, летальный исход, тяжкие травмы и/или материальный ущерб.

  Прежде чем вставить или удалить модуль из центральной стойки всегда следует отключать питание CPU и центральной стойки и придерживаться соответствующих мер безопасности.

- Карта памяти SIMATIC может вставляться или извлекаться при включенном CPU. Однако вставка или извлечение карты памяти в рабочем состоянии RUN вызывает переход CPU в состояние STOP.

  ВНИМАНИЕ

  Риски, связанные с извлечением карты памяти из CPU в режиме RUN

  При вставке или извлечении карты памяти в рабочем состоянии RUN происходит переход CPU в состояние STOP, что может вызвать неполадки на управляемых устройствах или в управляемом процессе.

  Каждый раз при вставке или извлечении карты памяти CPU сразу же переходит в рабочее состояние STOP. Прежде, чем вставить или извлечь карту памяти, следует убедиться, что CPU не находится в фазе активного управления машиной или процессом. Всегда следует оборудовать цепь АВАРИЙНОГО ОТКЛЮЧЕНИЯ для приложения или процесса.

- Если модуль вставляется или удаляется на стойке распределенного вода/вывода (AS-i, PROFINET или PROFIBUS), когда CPU находится в режиме RUN, CPU генерирует запись в диагностический буфер, исполняет OB вставки или удаления модулей, если он существует, и, по умолчанию, остается в рабочем состоянии RUN.

Обновление образов процесса и частичных областей отображения процесса

CPU обновляет локальные цифровые и аналоговые входы и выходы синхронно с циклом, используя внутреннюю область памяти, называемую образом процесса. Образ процесса содержит мгновенный снимок физических входов и выходов (физические каналы ввода/вывода CPU, сигнальных плат и сигнальных модулей).

Можно настроить I/O таким образом, что они будут обновляться в каждом цикле в образе процесса или при особом прерывающем событии. I/O могут быть настроены и так, что они будут исключены из обновления образа процесса. Пример: Процессу определенные значения данных могут понадобиться только при событии, напр., аппаратном прерывании. Назначая обновление образа процесса для этих I/O частичной области

отображения процесса, которая в свою очередь согласуется с OB аппаратного прерывания, удается избежать излишнего обновления этих значения данных со стороны CPU в каждом цикле, так как процессу не требуется непрерывное обновление.

Для I/O, которые обновляются в каждом цикле, CPU выполняется следующие задачи в каждом цикле:

- CPU записывает выходы из образа процессов выходов в физические выходы.
- CPU читает физические входы непосредственно перед выполнением программы пользователя и хранит входные значения в образе процесса входов. Таким образом, эти значения остаются неизменными при выполнении пользовательских инструкций.
- CPU обрабатывает логику пользовательских инструкций и обновляет значения на выходах в области образа процесса выходов вместо записи в фактические физические выходы.
  Этот процесс обеспечивает согласованную логику при выполнении пользовательских инструкций в соответствующем цикле и предотвращает мерцание физических выходов, выражающееся в многократном изменении состояния в образе процесса выходов.

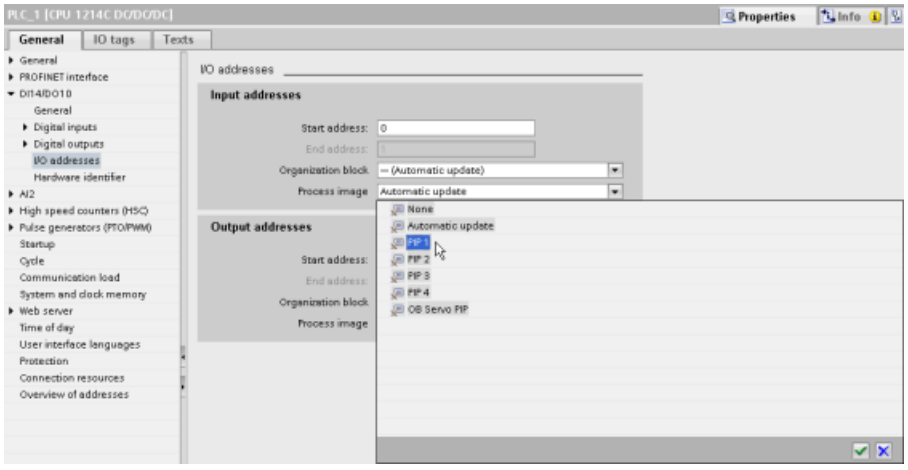
Для выбора, будет ли процесс обновлять I/O автоматически в каждом цикле или при возникновения событий, S7-1200 предоставляет пять частичных областей отображения процесса. Первая частичная область отображения процесса, PIP0, предназначена для I/O, обновляемых автоматически в каждом цикле. Это назначение по умолчанию. Оставшимся четырем частичным областям отображения процесса, PIP1, PIP2, PIP3 и PIP4, можно назначить обновление образа процесса при различных прерывающих событиях. I/O назначаются частичным областям отображения процесса в конфигурации устройства, частичные области отображения процесса в свою очередь назначаются прерывающим событиям при создании OB прерываний (Страница 183) или редактировании свойств OB (Страница 183).

При вставке модуля в просмотр устройств STEP 7 по умолчанию устанавливает для обновления образа процесса опцию "Автоматическое обновление". Для I/O, для которых заданно "Автоматическое обновление", CPU обрабатывает обмен данными между модулем и образом процесса автоматически в каждом цикле.

Для назначения цифровых или аналоговых I/O частичной области отображения процесса или для исключения I/O из обновления образа процесса, следует действовать следующим образом:

1. Открыть для соответствующего устройства вкладку "Свойства" в конфигурации устройства.
2. При необходимости раскрыть пункты в "Общее", чтобы показать нужные I/O.
3. Выбрать "Адреса I/O".

4. Как опцию выбрать определенный OB из выпадающего списка "Организационный блок".
5. Изменить через выпадающий список "Образ процесса" установку "Автоматическое обновление" на "PIP1", "PIP2", "PIP3", "PIP4" или "Нет". Выбор "Нет" означает, что чтение и запись этих I/O возможны только через прямые инструкции. Для повторного включения входов и выходов в автоматическое обновление образа процесса, снова установить для этой опции "Автоматическое обновление".



При выполнении инструкции возможно прямое чтение значений физических входов и прямая запись значений на физические выходы. При прямом чтении происходит обращение к текущему состоянию физического входа. Образ процесса входов независимо от того, сконфигурирован ли вход для сохранения в образец процесса, не обновляется. При прямой записи на физический выход обновляется как образ процесса выходов (если выход сконфигурирован для сохранения в образец процесса), так и физический выход. Добавить окончание ":P" к адресу I/O, чтобы программа обращалась к данным ввода/вывода напрямую через физический вход или выход, а не через образ процесса.

Примечание

Использование частичных областей отображения процесса

Если одному из частичных областей отображения процесса от TPA1 до TPA4 назначаются несколько I/O, но частичной области отображения процесса не назначается OB, то CPU никогда не обновляет эти I/O в образце процесса. Назначение I/O для PIP без соответствующего назначения OB равнозначно установке образа процесса для обновления "Нет". Можно прочитать I/O с помощью прямой инструкции чтения из физического I/O или записать в физический I/O с помощью прямой инструкции записи. CPU не обновляет образ процесса.

CPU поддерживает распределенную периферию для PROFINET, PROFIBUS и AS-i сетей (Страница 805).

5.1.1 Рабочие состояния CPU

У CPU есть три рабочих состояния: Рабочее состояние STOP, рабочее состояние STARTUP и рабочее состояние RUN. Светодиоды состояния на лицевой стороне CPU показывают текущее рабочее состояние.

- В рабочем состоянии STOP модуль CPU не выполняет программу. Можно загрузить проект в CPU.
- В рабочем состоянии STARTUP пусковые OB (при наличии) выполняются однократно. В этом рабочем состоянии CPU не обрабатывает события прерывания процесс.
- В рабочем состоянии RUN OB программного цикла выполняются в повторяющемся режиме. В любой рабочего состояния RUN могут возникнуть прерывающие события, вызывающие выполнение соответствующих OB прерывающих событий.
  Можно загрузить части проекта в рабочем состоянии RUN (Страница 1410).

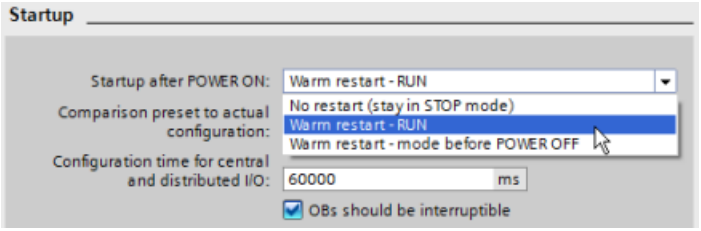
Модуль CPU поддерживает горячий пуск для перехода в рабочее состояние RUN. Восстановление исходного состояния памяти при горячем пуске не выполняется. Модуль CPU при горячем пуске инициализирует все не сохраняющиеся системные и пользовательские данные и буферизирует значения всех сохраняющихся данных пользователя.

При восстановлении исходного состояния памяти стирается содержание рабочей память, а также все сохраняющиеся и не сохраняющиеся области памяти, загрузочная память копируется в рабочую память и выходы переводятся в сконфигурированное состояние "Реакция на CPU-STOP". Диагностический буфер и постоянные значения для IP-адреса при восстановлении исходного состояния памяти сохраняются.

Можно сконфигурировать установку "Пуск после СЕТЬ ВКЛ" модуля CPU. Эта установка находится в конфигурации устройства CPU в "Пуск". При включении CPU выполняет ряд диагностических проверок с последующей инициализацией системы. Во время инициализации системы CPU очищает всю несохраняемую область памяти меркеров и сбрасывает все несохраняемое содержимое DB на начальные значения из загрузочной памяти. CPU буферизирует сохраняемую область памяти меркеров и сохраняемое содержимое DB и затем переходит в соответствующее рабочее состояние.

Определенные ошибки препятствуют переходу CPU в рабочее состояние RUN. CPU поддерживает следующие опции конфигурации:

- Без перезапуска (оставаться в состоянии STOP)
- Горячий пуск - RUN
- Горячий пуск - режим работы до СЕТЬ ВЫКЛ

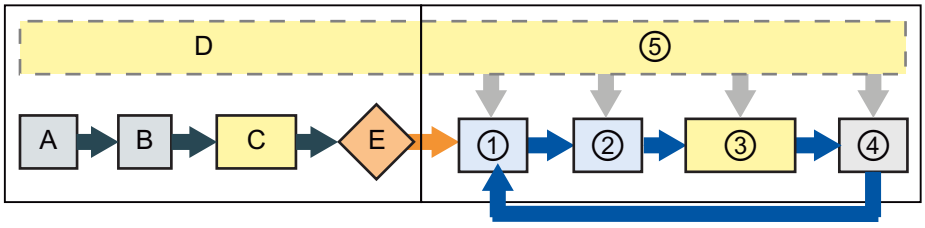


ВНИМАНИЕ
Поддающиеся исправлению ошибки могут вызвать переход CPU в рабочее состояние STOP. Из-за поддающихся исправлению ошибок CPU может перейти в рабочее состояние STOP, напр.: • Неисправность съемного сигнального модуля • Временные ошибки, такие как нарушение электропитания или непредвиденное событие при запуске. Такие ситуации могут стать причиной материального ущерба. Если для CPU была сконфигурирована установка "Горячий пуск - режим работы до СЕТЬ ВЫКЛ", то CPU переходит в рабочее состояние, в котором он находился до исчезновения напряжения питания или ошибки. Если CPU в момент исчезновения напряжения питания или ошибки находился в рабочем состоянии STOP, то при запуске он возвращается в состояние STOP. CPU остается в рабочем состоянии STOP, пока не получит команду на переход в режим RUN. Если CPU в момент исчезновения напряжения питания или ошибки находился в рабочем состоянии RUN, то при запуске он возвращается в состояние RUN. Модуль CPU возвращается в режим RUN, если он не обнаруживает ошибок, которые препятствовали бы переходу в RUN. Модули CPU, которые должны работать независимо от соединения STEP 7, должны быть сконфигурированы на "Горячий пуск - RUN". Здесь CPU при следующем запуске может вернуться в рабочее состояние RUN.

Для изменения текущего рабочего состояния можно использовать команды "STOP" и "RUN" (Страница 1396) в интерактивном инструментарии ПО для программирования. Кроме этого, можно вставить инструкцию STP (Страница 319) в программу, чтобы

перевести CPU в рабочее состояние STOP. Это позволяет прервать выполнение программы в зависимости от ее логики.

- В рабочем состоянии STOP модуль CPU обрабатывает коммуникационные запросы (по необходимости) и выполняет самодиагностику. CPU не выполняет программу пользователя. Не выполняется автоматическое обновление образа процесса.
- В рабочих состояниях STARTUP и RUN модуль CPU выполняет представленные на рисунке ниже задачи.



STARTUP

A	Состояние физических входов копируется в область памяти I
B	Инициализация области памяти Q с нулевым, последним или замещающим значением в зависимости от конфигурации. Выходы PB, PN и AS-i устанавливаются на ноль.
C	Не сохраняемые области памяти меркеров и блоки данных инициализируются с их начальным значением, а сконфигурированные события циклических прерываний и события по времени активируются. Пусковые OB выполняются.
D	Все прерывающие события сохраняются в лист ожидания для обработки в рабочем состоянии RUN
E	Активируется запись область памяти Q на физические выходы

RUN

①	Область памяти Q записывается на физические выходы
②	Состояние физических входов копируется в область памяти I
③	OB программного цикла выполняются
④	Выполняется самодиагностика
⑤	Прерывания и коммуникация обрабатываются на всех этапах цикла

Примечание

Коммуникация, включая коммуникацию HMI, может прерывать только OB программного цикла.

Процесс запуска (STARTUP)

Всякий раз, когда рабочий режим меняется со STOP на RUN, CPU очищает входные данные образа процесса, инициализирует выходные данные образа процесса и обрабатывает пусковые OB. Любой доступ по чтению к входным данным образа процесса с помощью инструкций в пусковых OB дает ноль, а не текущее значение физического входа. Поэтому для считывания текущего состояния физического входа в рабочем состоянии STARTUP необходимо выполнить прямое считывание входа. Пусковой OB и соответствующие FC и FB выполняются после. Если существует более одного пускового OB, то CPU обрабатывает каждый OB по порядку в соответствии с номером OB, начиная с младшего номера.

Каждый пусковой OB включает в себя информацию о запуске, которая помогает определить достоверность сохраняющихся данных и часов реального времени. В пусковых OB можно запрограммировать инструкции для анализа пусковых значений и принятия соответствующих мер. Следующие пусковые ячейки поддерживаются пусковыми OB:

Таблица 5-1 Поддерживаемые пусковым OB ячейки

Вход	Тип данных	Описание
LostRetentive	Bool	Этот бит принимает значение "истина", если области памяти сохраняемых данных были потеряны.
LostRTC	Bool	Этот бит принимает значение "истина", если часы реального времени были потеряны.

CPU также выполняет следующие задачи во время процедуры запуска.

- Прерывания на этапе запуска помещаются в лист ожидания и не обрабатываются
- На этапе запуска время цикла не контролируется
- Конфигурация модулей HSC (модуль быстрого счета), широтно-импульсной модуляции (ШИМ) и коммуникации точка-точка может быть изменена при запуске
- Фактическая работа модулей HSC, PWM и PTP происходит только в рабочем состоянии RUN

После выполнения пусковых OB, модуль CPU переходит в рабочее состояние RUN и обрабатывает задачи управления в циклическом режиме.

5.1.2 Обработка цикла в рабочем состоянии RUN

В каждом цикле CPU записывает выходные данные, считывает входные данные, выполняет программу пользователя, обновляет коммуникационные модули и отвечает на прерывающие события пользователя и коммуникационные запросы. Коммуникационные запросы обрабатываются регулярно в течение цикла.

Эти действия (за исключением прерывающих событий пользователя) обрабатываются регулярно и последовательно. Активированные прерывающие события пользователя выполняются в соответствии с их приоритетом в том порядке, в котором они происходят. Для прерывающего событий CPU считывает входные данные, выполняет OB, а затем записывает выходные данные, используя соответствующую частичную область отображения процесса (PIP), если это применимо.

Система гарантирует, что цикл будет обработан в течение максимального времени цикла, иначе создается ошибка по времени.

- Каждый цикл начинается с опроса текущих значений цифровых и аналоговых выходов в образе процесса и последующей записи их на физические выходы модулей CPU, SB и SM, настроенных для автоматического обновления I/O (конфигурация по умолчанию). При обращении инструкции к физическому выходу, обновляется как выход в образе процесса, так и сам физический выход.
- Цикл продолжается путем считывания текущих значений цифровых и аналоговых входов из CPU, SB и SM, настроенных на автоматическое обновление I/O (конфигурация по умолчанию) с последующей записью этих значений в образ процесса. Когда инструкция обращается к физическому входу, то значение физического входа изменяется, но вход в образе процесса не обновляется.
- После считывания входных данных программа пользователя выполняется от первой до последней инструкции. Здесь включены все OB программных циклов и все связанные с ними FC и FB. OB программных циклов выполняются по порядку в соответствии с номером OB, начиная с OB с младшим номером.

Обработка коммуникации происходит периодически на протяжении всего цикла и может прерывать выполнение программы пользователя.

Самодиагностика включает в себя периодические проверки состояния системы и опрос состояния модулей ввода-вывода.

Прерывания могут возникать в любой части цикла и управляться событиями. При возникновении события CPU прерывает цикл и вызывает OB для обработки этого события. После того, как OB завершает обработку события, CPU возобновляет выполнение программы пользователя в точке прерывания.

5.1.3 Организационные блоки (OB)

OB управляют выполнением программы пользователя. Определенные события в CPU запускают выполнение организационного блока. OB не могут вызывать друг друга. Также они не могут быть вызваны из FC или FB. Только такое событие, как диагностическое прерывание или временной интервал, может начать выполнение OB в CPU.
CPU обрабатывает OB в соответствии с их классами приоритетов, причем OB с более высоким приоритетом выполняется перед OB с более низким приоритетом. Самый низкий класс приоритетов - 1 (для основного программного цикла), а самый высокий класс приоритетов - 26.

5.1.3.1 OB программного цикла

OB программного цикла обрабатываются циклически, если CPU находится в режиме RUN. Основным блоком программы является OB программного цикла. Он содержит инструкции для управления программой, и из него вызываются дополнительные пользовательские блоки. Можно создать несколько OB программного цикла, которые выполняются CPU в порядке их номеров. Блоком по умолчанию является блок Main (OB 1).

События программного цикла

Событие программного цикла происходит один раз в течение каждого цикла программы. Во время программного цикла процессор записывает выходные данные, считывает входные данные и выполняет OB программного цикла. Событие программного цикла является обязательным и всегда активировано. Возможно, что для события программного цикла нет OB программного цикла или есть несколько OB. После того, как происходит событие программного цикла, CPU выполняет OB программного цикла с младшим номером (как правило, это OB 1). CPU выполняет другие OB программного цикла последовательно (в порядке их номеров) в ходе программного цикла. Программа выполняется циклически так, что событие программного цикла происходит в следующие моменты времени:

- После завершения выполнения последнего пускового OB
- После завершения выполнения последнего OB программного цикла

Таблица 5-2 Стартовая информация для OB программного цикла

Вход	Тип данных	Описание
Initial_Call	Bool	Значение "истина" при первом вызове OB
Remanence	Bool	Значение "истина" при наличии сохраняющихся данных

5.1.3.2 Пусковой OB

Пусковые OB выполняются один раз, когда режим работы процессора меняется из STOP на RUN, при запуске в рабочем состоянии RUN и при команде перехода из STOP к RUN. После начинается выполнение программного цикла.

Пусковые события

Пусковое событие происходит один раз при переходе из STOP в RUN и является командой для CPU на выполнение пусковых OB. Можно сконфигурировать несколько OB для пускового события. Пусковые OB выполняются по номерам.

Таблица 5-3 Стартовая информация для пускового OB

Вход	Тип данных	Описание
LostRetentive	Bool	Значение "истина", если сохраняемые данные были потеряны
LostRTC	Bool	Значение "истина", если дата и время были потеряны

5.1.3.3 OB прерывания по задержке времени

OB прерывания по задержке времени выполняются по истечении сконфигурированной задержки времени.

События прерывания по задержке времени

События прерывания по задержке времени это события, наступление которых согласно конфигурации должно произойти после того, как истекло заданное время задержки. Время задержки назначается с помощью инструкции SRT_DINT. События задержки прерывают цикл программы, чтобы выполнить соответствующий OB прерывания по задержке времени. Одному событию задержки может быть назначен только один OB прерывания по задержке времени. CPU поддерживает четыре события задержки.

Таблица 5-4 Стартовая информация для OB прерывания по задержке времени

Вход	Тип данных	Описание
Sign	Word	Идентификатор, передаваемый в триггерный вызов SRT_DINT

5.1.3.4 OB циклических прерываний

OB циклических прерываний выполняются с заданным интервалом. Можно сконфигурировать до четырех событий циклических прерываний, при этом каждому событию соответствует один OB.

События циклических прерываний

События циклических прерываний позволяют настроить выполнение OB прерывания в заданное время цикла. Начальное время цикла устанавливается при создании OB циклических прерываний. Событие циклического прерывания останавливает цикл программы и выполняет соответствующий OB циклических прерываний. Следует помнить, что событие циклического прерывания имеет более высокий класс приоритетов, чем событие программного цикла.

Одному событию циклического прерывания может быть назначен только один OB циклических прерываний.

Можно назначить фазовый сдвиг каждому циклическому прерыванию, чтобы выполнение циклических прерываний могло быть смещено друг относительно друга на величину фазового сдвига. Например, при наличии события циклического прерывания с интервалом 5 мс и еще одного события циклического прерывания с интервалом 10 мс, каждые 10 миллисекунд оба события будут происходить в один и тот же момент времени. Если назначить фазовый сдвиг для события с интервалом 5 мс на 1-4 мс и на 0 мс для события с интервалом 10 мс, то эти два события никогда не наступят одновременно.

Фазовый сдвиг по умолчанию равен 0. Для изменения начального фазового сдвига или времени для циклического события, выполнить следующие действия:

1. Кликнуть правой кнопкой мыши по OB циклических прерываний в дереве проекта.
2. Выбрать в контекстном меню пункт "Свойства".
3. Нажать кнопку "Циклическое прерывание" в диалоговом окне "Циклическое прерывание [OB 30]" и ввести новые начальные значения.

Максимальный фазовый сдвиг составляет 6000 мс (6 секунд) или максимальное время цикла, в зависимости от того, что меньше.

Также можно запрашивать и изменять время цикла и фазовый сдвиг из программы. Для этого служат инструкции "Запрос параметров циклического прерывания" (QRY_CINT) и

"Установка параметров циклического прерывания" (SET_CINT). Введенные с помощью инструкции SET_CINT значения для времени цикла и фазового сдвига не сохраняются при выключении или переходе в режим STOP; при следующем включении или возврате в режим RUN снова используются исходные значения. CPU поддерживает в общей сложности четыре события циклических прерываний.

5.1.3.5 OB аппаратных прерываний

OB аппаратных прерываний выполняются при возникновении соответствующего аппаратного события. OB аппаратных прерываний останавливает нормальное выполнение программы в ответ на сигнал от аппаратного события.

События аппаратных прерываний

Изменения в аппаратном обеспечении, такие как передний или задний фронт на входе или событие на HSC (High Speed Counter), запускают события аппаратных прерываний. Система S7-1200 поддерживает один OB прерывания для каждого события аппаратного прерывания. Можно активировать аппаратные события в конфигурации устройства и назначить событию OB в конфигурации устройства или с помощью инструкции ATTACH в программе пользователя. CPU поддерживает несколько событий аппаратных прерываний. Какие события доступны, зависит от модели CPU и числа входов.

Для событий аппаратных прерываний действуют следующие предельные значения:

Фронты:

- События переднего фронта: максимум 16
- События заднего фронта: максимум 16

HSC события:

- CV=PV: максимум 6
- Изменение направления: максимум 6
- Внешний сброс: максимум 6

Таблица 5-5 Стартовая информация для OB аппаратных прерываний

Вход	Тип данных	Описание
LADDR	HW_IO	Аппаратный идентификатор модуля, вызвавшего аппаратное прерывание.
USI	WORD	Идентификатор структуры пользователя (16#0001 до 16#FFFF), зарезервировано для будущего использования
IChannel	USINT	Номер канала, запустившего прерывание
EventType	BYTE	Идентификатор ориентированного на модуль типа события, вызывающего прерывание, напр., переднего или заднего фронта

Биты в EventType зависят от запускающего модуля. См. ниже:

Модуль/ субмодуль	Значение	Событие процесса
Встроенные I/O на CPU или SB	16#0	Передний фронт
	16#1	Задний фронт
HSC	16#0	HSC CV=RV1
	16#1	HSC изменение направления
	16#2	HSC сброс
	16#3	HSC CV=RV2

5.1.3.6 OB прерывания по ошибке времени

OB прерывания по ошибке времени (OB 80), если он сконфигурирован, выполняется тогда, когда происходит превышение максимального времени цикла, либо происходит событие ошибки по времени. Срабатывание и выполнение OB прерывает обычное циклическое выполнение программы или любого другого OB события.

При возникновении любого из этих событий создается запись в диагностическом буфере, описывающая данное событие. Запись в диагностический буфер генерируется независимо от наличия OB прерывания по ошибке времени.

События прерывания по ошибке времени

Возникновение любой ошибки по времени приводит к событию ошибки по времени:

- Превышение максимального времени цикла
  Ошибка "Превышение максимального времени цикла" возникает тогда, когда программный цикл не завершается за указанное максимальное время цикла. Глава"Контроль и конфигурирование времени цикла" (Страница 107) содержит подробную информацию о максимальном времени цикла, настройке максимального времени цикла в свойствах CPU и сбросе счетчика циклов.
- CPU не может запустить запрошенный OB, так до того, как CPU завершил обработку первого OB прерывания, было запущено второе прерывание (циклическое прерывание или прерывание по задержке времени)
- Переполнение листа ожидания
  Ошибка "Переполнение листа ожидания" возникает тогда, когда прерывания возникают чаще, чем CPU успевает их обрабатывать. Число активных событий ограничивается CPU, при этом каждому типу события назначается собственный лист ожидания. Если возникает событие, а соответствующий лист ожидания заполнен, то CPU создает событие ошибки по времени.

Все события ошибок по времени вызывают выполнение OB прерывания по ошибке времени (при наличии такового). В случае отсутствия OB прерывания по ошибке времени процессор переходит в режим STOP.

Путем выполнения инструкции RE_TRIGR (Страница 318) для перезапуска контроля времени цикла программа пользователя может продлить время выполнения цикла программы до десяти раз от сконфигурированного максимального времени цикла. Однако, если ошибка "Максимальное время цикла превышено" возникает дважды в

одном цикле программы, не сбрасывая время цикла, то CPU переходит в состояние STOP, независимо от наличия OB прерывания по ошибки времени. Более подробная информация содержится в разделе "Контроль времени цикла" в справочнике по системе S7-1200 (Страница 107).

OB прерывания по ошибке времени включает в себя стартовую информацию, с помощью которой можно определить, какое событие и какой OB произвели ошибку по времени. В OB можно запрограммировать инструкции для анализа этих пусковых значений и принятия соответствующих мер.

Таблица 5-6 Стартовая информация для OB по ошибке времени (OB 80)

Вход	Тип данных	Описание
fault_id	BYTE	16#01 - Превышено максимальное время цикла 16#02 - Не удается запустить необходимый OB 16#07 и 16#09 - Переполнение листа ожидания
csg_OBnr	OB_ANY	Номер OB, который выполнялся в момент возникновения ошибки
csg_prio	UINT	Приоритет OB, вызвавшего ошибку

Для вставки OB прерывания по ошибке времени в проект, необходимо добавить прерывание по ошибке времени, дважды кликнув по "Добавить новый блок" в разделе "Программные блоки" в дереве проекта, затем выбрать "Организационный блок" и "Ошибка по времени".

Приоритет нового CPU V4. 0 составляет 22. При замене CPU V3.0 на CPU V4.0 (Страница 1655) приоритет равен 26, т.е. это приоритет, который действовал для V3.0. В обоих случаях поле приоритетов редактируется и можно выбрать для приоритета любое значение в диапазоне от 22 до 26.

5.1.3.7 OB диагностического прерывания

OB диагностического прерывания выполняется, когда CPU обнаруживает диагностическую ошибку или если модуль с поддержкой диагностики распознает ошибку и для модуля активируется диагностическое прерывание. OB диагностического прерывания останавливает нормальное выполнение программы. Если при обнаружении ошибки такого типа CPU сразу же должен перейти в состояние STOP, то можно вставить в OB диагностического прерывания инструкцию STP.

Если не вставить OB диагностического прерывания в программу, то CPU игнорирует ошибку и остается в режиме RUN.

События диагностических ошибок

Аналоговые (локальные), PROFINET-, PROFIBUS- и некоторые цифровые (локальные) устройства могут обнаруживать и сигнализировать диагностические ошибки. Возникновение или исчезновение любой из диагностических ошибок вызывает событие диагностической ошибки. Поддерживаются следующие диагностические ошибки:

- Нет напряжения
- Выход за верхний предел
- Выход за нижний предел

- Обрыв провода
- Короткое замыкание

События диагностических ошибок запускают выполнение OB диагностического прерывания (OB 82) при его наличии. Если он отсутствует, то CPU игнорирует ошибку.

Для вставки OB диагностического прерывания в проект необходимо добавить диагностическое прерывание. Для этого в дереве проекта выполнить двойной клик по "Добавить новый блок" в "Программных блоках" и выбрать "Организационный блок" и "Диагностическое прерывание".

Примечание

Диагностические ошибки для многоканальных локальных аналоговых устройств (ввод/вывод, терморезисторы и термопары)

OB диагностического прерывания может одновременно обрабатывать только одну диагностическую ошибку одного канала соответственно.

Если в двух каналах многоканального устройства возникают ошибки, то вторая ошибка запускает OB диагностического прерывания только при следующих условиях: Ошибка первого канала устраняется, выполнение OB диагностического прерывания, запущенное первой ошибкой, завершено, а вторая ошибка все еще активна.

OB диагностического прерывания содержит стартовую информацию, которая помогает определить, связано ли событие с возникновением или исчезновением ошибки, а также устройство и канал, которые сигнализировали ошибку. В OB диагностического прерывания можно запрограммировать инструкции для анализа этих пусковых значений и принятия соответствующих мер.

Примечание

Стартовая информация OB диагностического прерывания относится к субмодулю в целом, если нет активного диагностического события

В V3.0 стартовая информация исчезающего события диагностической ошибки всегда указывала на источник события. В V4.0, если исчезающее событие исключает активную диагностику для субмодуля, то стартовая информация все равно ссылается на субмодуль в целом (16#8000), даже если источником события был определенный канал.

Пример: Обрыв провода вызывает диагностическую ошибку в канале 2, ошибка исправляется, событие диагностической ошибки удаляется, но стартовая информация все равно ссылается не на канал 2, а на субмодуль (16#8000).

Таблица 5-7 Стартовая информация для OB диагностического прерывания

Вход	Тип данных	Описание
IOstate	WORD	Состояние ввода / вывода устройства: • Бит 0 = 1, если конфигурация правильная, и Бит 0 = 0, если конфигурация уже неправильная. • Бит 4 = 1, если имеется ошибка (пример: обрыв провода). (Бит 4 = 0, если не ошибок.) • Бит 5 = 1, если конфигурация **не**правильная, и бит 5 = 0, если конфигурация снова правильная. • Бит 7 = 1, если имеется ошибка доступа к I/O. Аппаратный идентификатор I/O с ошибкой доступа см. в LADDR. (Бит 6 = 0, если не ошибок.)
LADDR	HW_ANY	Аппаратный идентификатор устройства или функционального модуля, который сообщил об ошибке[1]
Channel	UINT	Номер канала
MultiError	BOOL	"ИСТИНА", если есть несколько ошибок

[1] Элемент LADDR содержит аппаратный идентификатор устройства или функционального модуля, который вывел ошибку. Аппаратный идентификатор назначается автоматически при вставке компонентов в просмотре устройств или сетевых соединений. Он отображается на вкладке "Константы" переменных (тегов) PLC. Имя аппаратному идентификатору также назначается автоматически. Эти записи на вкладке "Константы" переменных (тегов) PLC не могут быть изменены.

5.1.3.8 OB "Удаление или вставка модулей"

OB "Удаление или вставка модулей" выполняется, когда настроенный и не отключенный распределенный модуль ввода-вывода или субмодуль (PROFIBUS, PROFINET, AS-i) генерирует событие, связанное с вставкой или удалением модуля.

Событие "Удаление или вставка модулей"

При следующих условиях создается событие "Удаление или вставка модулей":

- Пользователь удаляет или вставляет сконфигурированный модуль
- Сконфигурированный модуль физически отсутствует в стойке расширения
- В стойке расширения находится несовместимый модуль, который не соответствует сконфигурированному модулю
- В стойке расширения находится совместимый со сконфигурированным модуль, но конфигурация не разрешает замену
- У модуля или субмодуля есть ошибки параметрирования

Если этот OB не был запрограммирован, то CPU остается в режиме RUN, когда наступает любое из этих состояния для сконфигурированного и не отключенного модуля распределенного ввода/вывода.

Независимо от того, был ли запрограммирован этот OB, CPU перейдет в состояние STOP, если любое из этих условий наступает для модуля в центральной стойке.

Таблица 5-8 Стартовая информация для OB "Удаление или вставка модулей"

Вход	Тип данных	Описание
LADDR	HW_IO	Аппаратный идентификатор
Event_Class	Byte	16#38: модуль вставлен 16#29: модуль удален
Fault_ID	Byte	ID ошибки

5.1.3.9 OB ошибки стойки или станции ("Rack or station failure OB")

OB ошибки стойки или станции ("Rack or station failure OB") выполняется, когда CPU обнаруживает ошибку или отказ коммуникации на распределенной стойке или станции.

Событие ошибки стойки или станции

CPU генерирует событие ошибки стойки или станции, когда обнаруживается одна из следующих ситуаций:

- Отказ системы ведущих устройств DP или системы PROFINET IO (для наступающего или уходящего события)
- Отказ ведомого устройства DP или IO устройства (для наступающего или уходящего события)
- Отказ субмодуля интеллектуальный прибора ввода-вывода PROFINET

Если этот OB не был запрограммирован, то CPU остается в режиме RUN, когда наступает любое из этих условий.

Таблица 5-9 Стартовая информация для OB ошибки стойки или станции

Вход	Тип данных	Описание
LADDR	HW_IO	Аппаратный идентификатор
Event_Class	Byte	16#38: уходящее событие 16#39: наступающее событие
Fault_ID	Byte	Идентификатор ошибки

5.1.3.10 OB прерывания по времени

OB прерывания по времени обрабатываются согласно сконфигурированным условиям. CPU поддерживает два OB прерывания по времени.

События прерывания по времени

Можно сконфигурировать событие прерывания по времени так, чтобы оно происходило в заданное время или циклически по следующим правилам:

- Ежеминутно: Прерывание происходит каждую минуту.
- Ежечасно: Прерывание происходит каждый час.
- Ежедневно: Прерывание происходит каждый день в определенное время (час и минута).
- Еженедельно: Прерывание происходит каждую неделю в определенное время в определенный день недели (например, каждый вторник в 4:30 пополудни).
- Ежемесячно: Прерывание происходит каждый месяц в определенное время в определенный день месяца. Для дня следует выбрать число в диапазоне от 1 до 28.
- В конце месяца: Прерывание происходит в последний день каждого месяца в определенное время.
- Ежегодно: Прерывание происходит каждый год в указанную дату (месяц и день). Нельзя указать дату 29 февраля.

Таблица 5-10 Стартовая информация для OB прерывания по времени

Вход	Тип данных	Описание
CaughtUp	Bool	Вызов OB подхвачен, из-за перевода часов вперед
SecondTimes	Bool	Вызов OB запускается повторно, из-за перевода часов назад

5.1.3.11 OB состояния

OB состояния выполняются, если ведомое устройство DPV1 или PNIO вызывают прерывание по состоянию. Это может случится, если компонент (модуль или стойка) ведомого устройства DPV1 или PNIO изменяет свое рабочее состояние, например из RUN в STOP.

События изменения состояния

Для получения подробной информации о событиях, которые вызывают прерывание по состоянию, следует обратиться к документации изготовителя для ведомого устройства DPV1 или PNIO.

Таблица 5-11 Стартовая информация для OB состояния

Вход	Тип данных	Описание
LADDR	HW_IO	Аппаратный идентификатор
Slot	UInt	Номер слота
Specifier	Word	Описание прерывания

5.1.3.12 OB прерывания по обновлению

OB прерывания по обновлению выполняются, если ведомое устройство DPV1 или PNIO вызывают прерывание по обновлению.

События прерывания по обновлению

Для получения подробной информации о событиях, которые вызывают прерывание по обновлению, следует обратиться к документации изготовителя для ведомого устройства DPV1 или PNIO.

Таблица 5-12 Стартовая информация для OB прерывания по обновлению

Вход	Тип данных	Описание
LADDR	HW_IO	Аппаратный идентификатор
Slot	UInt	Номер слота
Specifier	Word	Описание прерывания

5.1.3.13 OB прерывания по профилю

OB прерывания по профилю выполняются, если ведомое устройство DPV1 или PNIO вызывают связанное с конкретным профилем прерывание.

События прерывания по профилю

Для получения подробной информации о событиях, которые вызывают прерывание по профилю, следует обратиться к документации изготовителя для ведомого устройства DPV1 или PNIO.

Таблица 5-13 Стартовая информация для OB прерывания по профилю

Вход	Тип данных	Описание
LADDR	HW_IO	Аппаратный идентификатор
Slot	UInt	Номер слота
Specifier	Word	Описание прерывания

5.1.3.14 OB MC-Servo и MC-Interpolator

STEP 7 генерирует защищенные по записи организационные блоки MC-Servo и MC-Interpolator автоматически при создании технологического объекта для управления перемещением и установки "Подключение аналогового привода" или "PROFIdrive" для интерфейса привода. Изменение свойств OB не требуется, как и прямое создание этого OB. CPU использует эти OB для регулирования. Дополнительная информация находится в информационной системе STEP 7.

5.1.3.15 MC-PreServo

Можно запрограммировать OB MC-PreServo таким образом, чтобы он содержал логику для программы STEP 7. В этом случае логика программы выполняется непосредственно перед обработкой OB MC-Servo.

События MC-PreServo

С помощью OB MC-PreServo сконфигурированная информация о цикле пользователя может считана за несколько микросекунд.

Таблица 5-14 Стартовая информация для OB MC-PreServo

Вход	Тип данных	Описание
Initial_Call	BOOL	"ИСТИНА" указывает, что первый вызов этого OB выполняется при переходе из STOP в RUN
PIP_Input	BOOL	"ИСТИНА" указывает на то, что соответствующий образ процесса входов имеет самую последнюю версию.
PIP_Output	BOOL	"ИСТИНА" указывает на то, что соответствующий образ процессов выходов был своевременно передан CPU на выходы после последнего цикла.
IO_System	USINT	Номер системы распределенного ввода-вывода, запускающей прерывание
Event_Count	INT	n: число выпавших циклов -1: неизвестное число выпавших циклов (напр., из-за изменения цикла)
Synchronous	BOOL	Зарезервировано
CycleTime	UDINT	Отображение сконфигурированного для OB MC-Servo цикла пользователя в микросекундах

5.1.3.16 MC-PostServo

Можно запрограммировать OB MC-PreServo таким образом, чтобы он содержал логику для программы STEP 7. В этом случае логика программы выполняется непосредственно после обработкой OB MC-Servo.

События MC-PostServo

С помощью OB MC-PostServo сконфигурированная информация о цикле пользователя может считана за несколько микросекунд.

Таблица 5-15 Стартовая информация для OB MC-PostServo

Вход	Тип данных	Описание
Initial_Call	BOOL	"ИСТИНА" указывает, что первый вызов этого OB выполняется при переходе из STOP в RUN
PIP_Input	BOOL	"ИСТИНА" указывает на то, что соответствующий образ процесса входов имеет самую последнюю версию.

Вход	Тип данных	Описание
PIP_Output	BOOL	"ИСТИНА" указывает на то, что соответствующий образ процессов выходов был своевременно передан CPU на выходы после последнего цикла.
IO_System	USINT	Номер системы распределенного ввода-вывода, запускающей прерывание
Event_Count	INT	n: число выпавших циклов -1: неизвестное число выпавших циклов (напр., из-за изменения цикла)
Synchronous	BOOL	Зарезервировано
CycleTime	UDINT	Отображение сконфигурированного для OB MC-Servo цикла пользователя в микросекундах

5.1.3.17 Приоритеты и лист ожидания для обработки событий

Обработка в CPU управляется событиями. Событие вызывает выполнение OB прерывания. Во время создания блока, во время конфигурации устройства, или с помощью инструкций ATTACH илиDETACH можно указать OB прерывания для события. Некоторые события, например, программные циклы или циклические события, происходят на регулярной основе. Другие события происходят лишь однократно, например событие запуска и события с временной задержкой. Некоторые события происходят, если его запускает оборудование, напр., фронт на входе или событие быстрого счета. Такие события, как диагностическая ошибка или ошибка по времени, возникают только при ошибках. Приоритеты событий и списки ожидания служат для определения порядка обработки для OB прерываний.

CPU обрабатывает события в порядке приоритета, где 1 - это самый низкий приоритет, а 26 - самый высокий приоритет. До V4.0 CPU S7-1200 каждый тип OB относился к определенному классу приоритетов (1 - 26). Начиная с V4.0, можно назначить класс приоритетов для каждого сконфигурированного OB. Класс приоритетов назначается в свойствах OB.

Прерываемые и непрерываемые режимы обработки

OB (Страница 91) обрабатываются в порядке приоритета событий, которые их вызывают. В Пусковых характеристиках конфигурации устройства CPU (Страница 167) можно настроить выполнение OB как прерываемое или не прерываемое. Следует помнить, что OB программного цикла всегда являются прерываемыми. Все другие OB могут быть настроены как прерываемые или не прерываемые.

Если выбирается прерываемый режим выполнения, то обработка OB прерывается, если событие с более высоким приоритетом происходит до завершения обработки этого OB, чтобы обеспечить выполнение OB с более высоким приоритетом. После выполняется событие с более высоким приоритетом, и по его завершении продолжается прерванный OB. Если во время выполнения прерываемого OB происходит несколько событий, то CPU обрабатывает эти события в порядке их приоритетов.

Если задается непрерываемый режим выполнения, то обработка OB выполняется до конца, независимо от любых других событий, которые возникают в течение этого времени.

Рассмотрим следующие два случая, когда прерывающие события запускают OB программного цикла и OB прерывания по задержке времени. В обоих случаях OB прерывания по задержке времени (OB 201) не назначен частичной области отображения процесса (Страница 83) и выполняется с приоритетом 4. OB программного цикла (OB 200) назначен частичной области отображения процесса PIP1 и выполняется с приоритетом 2. На рисунках ниже показана разница в обработке между режимами не прерываемого и прерываемого выполнения:

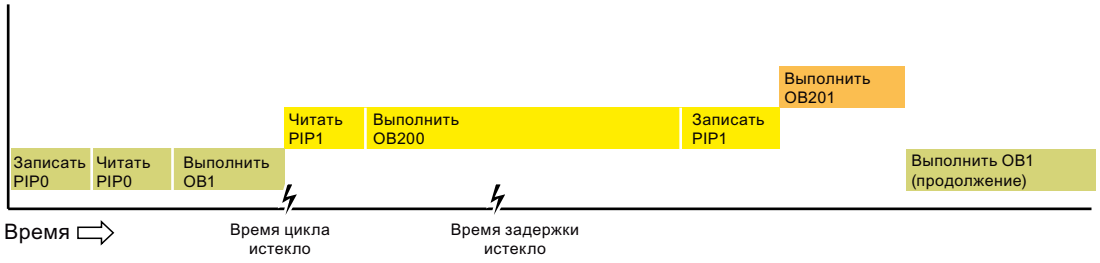


Рисунок 5-1 Сценарий 1: Непрерываемое выполнение OB

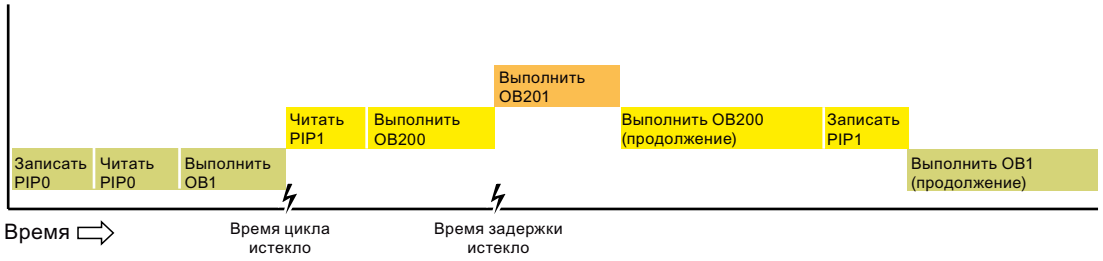


Рисунок 5-2 Сценарий 2: Прерываемое выполнение OB

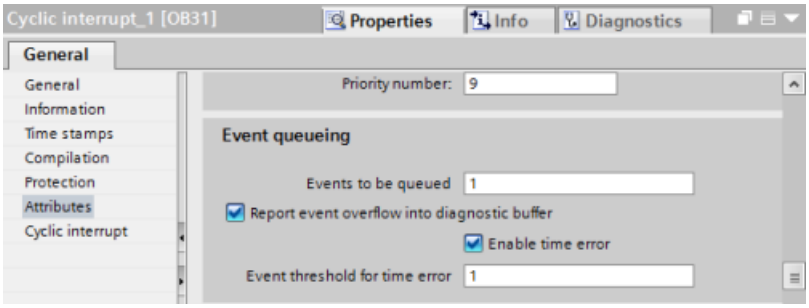
Примечание

Если выбирается непрерываемый режим выполнения OB, то OB по ошибке времени может прерывать только OB программного цикла. До V4.0 модуля S7-1200 CPU OB по ошибке времени мог прерывать любой исполняемый OB. Начиная от V4.0 необходимо сделать выполнение OB прерываемым, чтобы OB по ошибке времени (или иной OB с более высоким приоритетом) могли бы прерывать и другие исполняемые OB, а не только OB программного цикла.

Объяснение приоритетов и листа ожидания для обработки событий

Число активных событий из одного единственного источника ограничивается CPU, при этом каждому типу события назначается собственный лист ожидания. При достижении максимального количества активных событий определенного типа, следующее событие более не обрабатывается и теряется. С помощью OB прерывания по ошибке времени (Страница 95) можно реагировать на переполнение листа ожидания.

Следует помнить, что в STEP 7 можно сконфигурировать несколько специальных параметров для листа ожидания событий для OB циклических прерываний и OB прерывания по времени.



Дополнительная информация о поведении CPU при перегрузке и о листе ожидания для событий находится в информационной системе STEP 7.

У каждого события в CPU есть приоритет. События, как правило, обрабатываются CPU в последовательности их приоритетов (от высокого к низкому). События с одинаковым приоритетом обрабатываются CPU по принципу "первый пришел, первый обслужен".

Таблица 5-16 OB обработки событий

Событие	Разрешенное количество	Приоритет OB по умолчанию
Программный цикл	1 событие программного цикла Разрешается несколько OB	1[1]
Пуск	1 пусковое событие [1] Разрешается несколько OB	1[1]
Задержка	До 4 событий 1 OB на событие	OB 20: 3 OB 21: 4 OB 22: 5 OB 23: 6 OB 123 до OB 32767: 3
Циклическое прерывание	До 4 событий 1 OB на событие	OB 30: 8 OB 31: 9 OB 32: 10 OB 33: 11 OB 34: 12 OB 35: 13 OB 36: 14 OB 37: 16 OB 38: 17 OB 123 до OB 32767: 7
Аппаратное прерывание	До 50 событий аппаратных прерываний[2]	18
	1 OB на событие, но можно использовать тот же самый OB для нескольких событий	18

Событие	Разрешенное количество	Приоритет OB по умолчанию
Ошибка по времени	1 событие (только если сконфигурировано)[3]	22 или 26[4]
Диагностическая ошибка	1 событие (только если сконфигурировано)	5
Удаление или вставка модулей	1 событие	6
Ошибка стойки или станции	1 событие	6
Прерывание по времени	До 2 событий	2
Состояние	1 событие	4
Обновление	1 событие	4
Профиль	1 событие	4
MC-Servo	1 событие	25
MC-Interpolator	1 событие	24

[1] Пусковое событие и событие программного цикла никогда не происходят в одно и тоже время, потому что сначала должен быть завершен пуск, а лишь потом будет запущен программный цикл.

[2] Количество OB аппаратных прерываний может превышать 50, если используются инструкции DETACH и ATTACH.

[3] Можно сконфигурировать CPU таким образом, что при превышении максимального времени цикла он будет оставаться в режиме RUN, либо с помощью инструкции RE_TRIGR время цикла может быть сброшено. Но если максимальное время цикла будет превышено во второй раз за цикл, то CPU переходит в рабочее состояние STOP.

[4] Приоритет нового CPU V4. 0 или V4.1 равен 22. При замене CPU V3.0 на CPU V4.0 или V4.1 приоритет равен 26: т.е. это приоритет, который действовал для V3.0. В обоих случаях поле приоритетов редактируется и можно выбрать для приоритета любое значение в диапазоне от 22 до 26.

Дополнительная информация см. "Замена CPU V3.0 на CPU V4.x (Страница 1655)".

Кроме этого, CPU распознает и другие события, у которых нет соответствующего OB. В следующей таблице представлены эти события и соответствующие действия CPU:

Таблица 5-17 Дополнительные события

Событие	Описание	Действие CPU
Ошибка доступа к периферии	Ошибка при прямом чтении/записи I/O	При первом возникновении ошибки CPU создает запись в буфере диагностики и остается в режиме RUN. Причину ошибки можно запросить с помощью инструкции GET_ERROR_ID (Страница 320).
Ошибка максимального времени цикла	CPU превышает сконфигурированное время цикла во второй раз	CPU записывает ошибку в буфер диагностики и переходит в рабочее состояние STOP.

Событие	Описание	Действие CPU
Ошибка доступа к периферии	Ошибка ввода/вывода при обновлении образа процесса	При первом возникновении ошибки CPU создает запись в буфере диагностики и остается в режиме RUN.
Ошибка программирования	Ошибка при выполнении программы	• Если активирована локальная поблочная обработка ошибок, то система вносит причину ошибки в структуру ошибок. Причину ошибки можно запросить с помощью инструкции GET_ERROR_ID (Страница 320). • Если активирована глобальная обработка ошибок, то система вносит стартовое событие ошибки доступа в буфер диагностики и остается в режиме RUN.

Задержка

Задержка прерывания по событию (т.е. время от уведомления CPU о наступлении события до начала выполнения первой инструкции в OB для обработки события) составляет приблизительно 175 мкс, при условии, что OB программного цикл является единственной сервисной программой, активной на момент прерывающего события.

5.1.4 Контроль и конфигурирование времени цикла

Время цикла - это время, которое требуется операционной системе CPU, чтобы выполнить циклическую фазу рабочего состояния RUN. CPU предлагает два варианта контроля для времени цикла:

- Максимальное время цикла
- Минимальное время цикла

Контроль цикла начинается после завершения пускового события. Конфигурирование этой функции в CPU выполняется в "Конфигурации устройства> Время цикла".

CPU контролирует цикл и реагирует, если время цикла превышает максимальное сконфигурированное время цикла. CPU создает ошибку и реагирует следующим образом, если время цикла превышает максимальное сконфигурированное время цикла:

- Если в программе пользователя есть OB прерывания по ошибке времени (Страница 95), то он выполняется CPU.
- Если в программе пользователя не OB прерывания по ошибке времени, то событие ошибки по времени создает запись в буфере диагностики. После CPU переходит в рабочее состояние STOP.

Инструкция RE_TRIGR (Страница 318) (перезапуск контроля времени цикла) позволяет сбросить таймер для измерения времени цикла. Если истекшее время для выполнения текущего цикла программы меньше десятикратного установленного максимального времени цикла, то инструкция RE_TRIGR повторно запускает контроль времени цикла и

возвращается ENO = TRUE. Если нет, инструкция RE_TRIGR не перезапускает контроль времени цикла. Она возвращает ENO = FALSE.

Как правило, цикл выполняется настолько быстро, насколько это возможно, и следующий цикл начинается, как только текущий завершается. Но в зависимости от программы пользователя и задач коммуникации время цикла может меняться. Чтобы исключить такие колебания, CPU поддерживает опциональное минимальное время цикла. Если активировать эту опциональную функцию и задать минимальное время цикла в мс, то CPU после выполнения OB программного цикла ожидает, пока не истечет минимальное время цикла, прежде чем повторять программный цикл.

В случае, когда CPU завершает цикл за меньшее время, чем указанное минимальное время цикла, CPU тратит оставшееся время цикла на оперативную диагностику и/или обработку коммуникационных запросов.

Если цикл не завершается в течение заданного минимального времени цикла, то он выполняется до конца в штатном режиме (включая обработку коммуникации) и превышение минимального времени не вызывает какой-либо реакции системы. В следующей таблице определены диапазоны и значения по умолчанию для контроля времени цикла.

Таблица 5-18 Диапазон для времени цикла

Время цикла	Диапазон (мс)	По умолчанию
Максимальное время цикла[1]	От 1 до 6000	150 мс
Минимальное время цикла[2]	От 1 до максимального времени цикла	Отключено

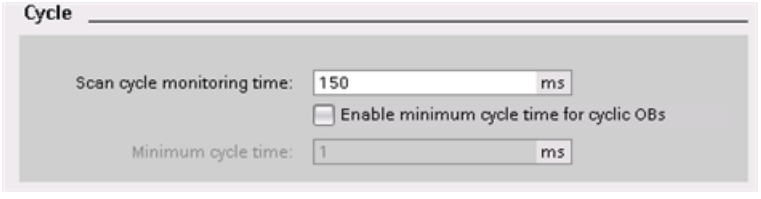
[1] Максимальное время цикла всегда активировано. Установить время цикла в диапазоне от 1 до 6000 мс. По умолчанию используется 150 мс.

[2] Минимальное время цикла это опция и по умолчанию оно отключено. При необходимости установить время цикла в диапазоне от 1 мс до максимального времени цикла.

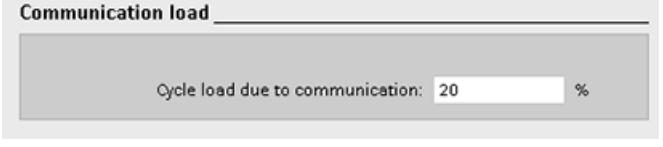
Конфигурирование времени цикла и коммуникационной нагрузки

В свойствах CPU в конфигурации устройств можно установить следующие параметры:

- Цикл: Здесь можно ввести максимальное время контроля цикла. Также можно также ввести / активировать минимальное время цикла.



- Коммуникационная нагрузка: Можно задать процент времени, которое будет выделено для коммуникационных задач.



Примечание

Приоритет коммуникации

Задачи коммуникации имеют приоритет 1. Так как 1 - это самый низкий приоритет, другие события CPU могут прерывать обработку коммуникации. Прерывания другими событиями могут отрицательно сказываться на обработке коммуникации в ходе цикла. Можно изменить процентное значение для "Коммуникационной нагрузки на цикл", чтобы увеличить долю цикла, предоставляемую для обработки коммуникации.

Подробная информация о цикле в "Контроль цикла" (Страница 107).

5.1.5 Память CPU

Управление памятью

CPU предоставляет следующие области памяти для программы пользователя, данных и конфигурации:

- Загружаемая память – это энергонезависимая память для программы пользователя, данных и конфигурации. При загрузке проекта в CPU он сначала сохраняется в загружаемой памяти. Эта память находится либо на карте памяти (если она имеется), либо в CPU. Данные в этой энергонезависимой области памяти сохраняется CPU и при отключении питания. Карта памяти поддерживает больший объем памяти, чем память, встроенная в CPU.
- Рабочая память – это энергозависимая память для некоторых элементов проекта пользователя во время исполнения программы пользователя. CPU копирует некоторые элементы проекта из загружаемой памяти в рабочую память. Эта энергозависимая область памяти теряется при отключении питания, а при возвращении питания CPU ее восстанавливает.
- Сохраняющая память - это энергонезависимая память для ограниченного количества значений рабочей памяти. CPU использует область сохраняющей памяти для сохранения значений выбранных адресов памяти пользователя при потере питания. Когда происходят отключение или перебой питания, CPU снова восстанавливает эти сохраняемые значения при запуске.

Для отображения использования памяти скомпилированным программным блоком, открыть контекстное меню кликом правой кнопкой мыши по блоку в папке "Программные блоки" дерева проекта STEP 7 и выбрать пункт "Ресурсы". В свойствах компиляции отображаются загружаемая память и рабочая память для скомпилированного блока.

Для отображения используемой памяти для CPU онлайн, дважды кликнуть по "Онлайн и диагностика" в STEP 7, раскрыть "Диагностика" и выбрать "Память".

Сохраняющая память

Можно избежать потери данных при выходе из строя питания, пометив некоторые данные как сохраняющиеся. В качестве сохраняющихся в CPU могут быть определены следующие данные:

- Меркеры (M): Можно определить размер сохраняющей памяти для меркеров в таблице переменных PLC или в списке назначений. Сохраняющая битовая память всегда начинается с MB0 и последовательно занимает указанное количество байтов. Указать это значение в таблице переменных PLC или в списке назначений, кликнув по значку "С сохранением" на панели инструментов. Указать количество M байтов для сохранения, начиная с MB0.
  Примечание: Можно показать список назначений для каждого блока, выбрав блок в папке "Программные блоки" и далее команду меню **Инструменты > Список назначений** .
- Переменные (теги) функционального блока (FB): Если FB имеет тип "оптимизированный доступ к блоку", то в редакторе интерфейса для этого FB имеется столбец "С сохранением". В этом столбце можно выбрать "С сохранением", "Без сохранения" или "Установить в IDB" отдельно для каждой переменной (тега). Размещая такой FB в программе, DB экземпляра, соответствующий FB, также включает этот столбец "С сохранением". Можно изменить сохраняемое состояние переменной (тега) в редакторе интерфейса DB экземпляра, только если в оптимизированном FB в поле "С сохранением" для переменной (тега) была выбрана опция "Установить в IDB".
  Если тип FB это **не** "оптимизированный доступ к блоку", то в редакторе интерфейса для этого FB нет столбца "С сохранением". Но размещая такой FB в программе следует учитывать, что DB экземпляра, соответствующий FB, все же включает в себя доступный для редактирования столбец "С сохранением". В этом случае выбор опции "С сохранением" для любой переменной (тега) приводит к выбору **всех** переменных (тегов). Аналогичным образом, отмена выбора опции "С сохранением" для любой переменной (тега) приводит к отмене выбора для **всех** переменных (тегов).
  Чтобы показать, имеет или FB оптимизированный доступ к блокам, следует открыть свойства FB и выбрать атрибуты.
- Переменные (теги) глобального блока данных: Поведение глобального DB в части назначения сохраняющего состояния аналогично поведению FB. В зависимости от настройки доступа к блоку можно определить сохраняющее состояние либо для отдельных, либо для всех переменных (тегов) глобального блока данных.
  - Если при создании DB выбрать "С оптимизацией", то можно установить сохраняющее состояние для каждой отдельной переменной (тега).
  - Если при создании DB выбирается "По умолчанию - совместимость с S7 - 300/400", то установка сохраняющее состояния применяется ко всем переменным (тегам) DB; т.е. либо все переменные (теги) являются сохраняющимися, либо ни одна переменная (тег) не является сохраняющимися.

CPU поддерживает в общей сложности 10240 байт сохраняющихся данных. Чтобы узнать число доступных байтов, кликнуть в таблице переменных PLC или в списке назначений на панели инструментов по значку "С сохранением". Хотя именно здесь задается сохраняющая область для битовой M памяти, но во второй строке указывается общая оставшаяся память, доступная для M и DB вместе. Следует помнить, что это значение

будет точным только после компиляции всех блоков данных с сохраняющимися переменными (тегами).

Примечание

Загрузка программы не удаляет и не изменяет значения в сохраняющей памяти. Для очистки сохраняющей памяти перед загрузкой следует сбросить CPU на заводские установки перед загрузкой программы.

5.1.5.1 Системная и тактовая память

В свойствах CPU можно активировать байты для "системной памяти" и "тактовой памяти". Логика программы может ссылаться на отдельные биты этих функций по их именам переменных (тегов).

- Можно назначить один байт в битовой M памяти для системной памяти. Байт системной памяти предоставляет следующие четыре бита, на которые программа пользователя может ссылаться через следующие имена переменных (тегов):
  - Первый цикл: Бит (имя переменной (тега) "FirstScan") по завершении пускового OB устанавливается на первый цикл на 1. (По завершении первого цикла бит "Первый цикл" устанавливается на 0.)
  - Диагностическое состояние изменилось: (Имя переменной (тега): "DiagStatusUpdate") после регистрации CPU диагностического события в течение одного цикла остается на 1. Так как CPU устанавливает бит "DiagStatusUpdate" только в конце первого выполнения OB программного цикла, программа пользователя не может определить, произошло ли диагностическое изменение во время выполнения пускового OB или первого выполнения OB программного цикла.
  - Всегда 1 (высокий уровень): Бит (имя переменной (тега) "AlwaysTRUE") всегда установлен на 1.
  - Всегда 0 (низкий уровень): Бит (имя переменной (тега) "AlwaysFALSE") всегда установлен на 0.
- Можно назначить один байт в битовой M памяти для тактовой памяти. Каждый бит этой сконфигурированной как тактовая память байта генерирует прямоугольный импульс. Байт тактовой памяти предлагает 8 различных частот, от 0,5 Гц (медленно) до 10 Гц (быстро). Можно использовать эти биты в качестве управляющих битов, в первую очередь в сочетании с инструкциями выделения фронта, для циклического запуска операций в программе пользователя.

CPU инициализирует эти байты при переходе из состояния STOP в режим STARTUP. Биты тактовой памяти в режимах работы STARTUP и RUN изменяются синхронно с тактом CPU.

⚠ ОСТОРОЖНО

Риски при перезаписи битов системной или тактовой памяти

Перезапись битов системной или тактовой памяти может повредить данные в этих функциях и привести к ошибкам в работе программы пользователя, что, в свою очередь, может стать причиной материального ущерба и травм персонала.

Поскольку тактовая и системная память являются не зарезервированной памятью в области меркеров, инструкции или коммуникации могут выполнить запись в эти ячейки и повредить данные.

Избегать записи данных в этим ячейки, чтобы гарантировать правильную работу этих функций, и всегда использовать схему аварийного отключения для процесса или машины.

Системная память конфигурирует байт с битами, которые становятся активными (значение = 1) при определенном событии.

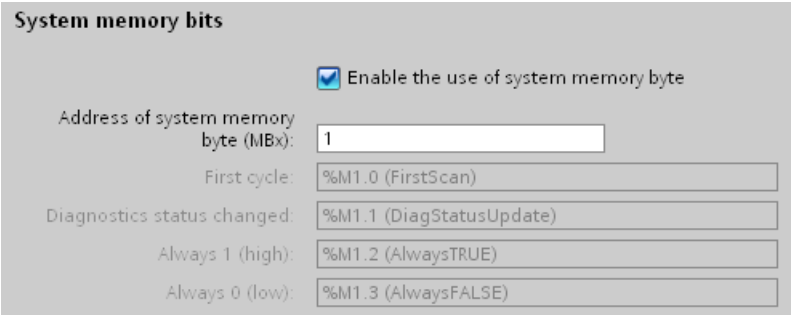


Таблица 5-19 Системная память

7	6	5	4	3	2	1	0
Зарезервировано Значение 0				Всегда отключено Значение 0	Всегда включено Значение 1	Индикатор состояния диагностики • 1: Изменение • 0: Без изменений	Индикатор первого цикла • 1: Первый цикл после запуска • 0: Не первый цикл

Тактовая память конфигурирует байт, включающий и выключающий отдельные биты через определенные интервалы. Каждый тактовый бит создает прямоугольный импульс в соответствующей битовой M памяти. Можно использовать эти биты в качестве управляющих битов, в первую очередь в сочетании с инструкциями выделения фронта, для циклического запуска операций в программе пользователя.

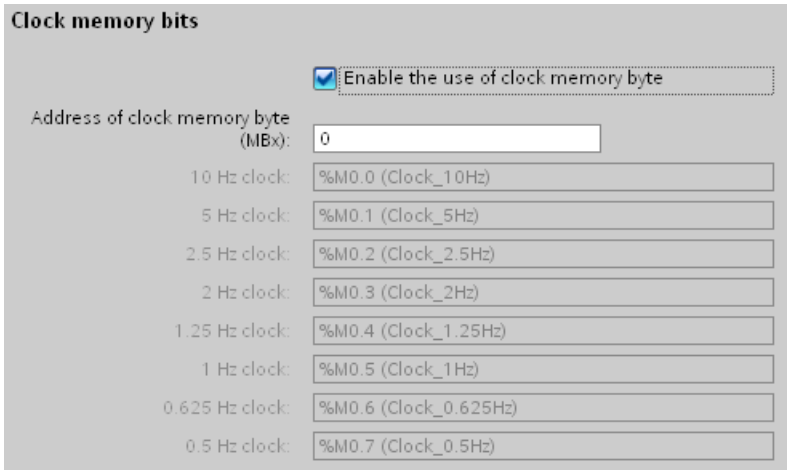


Таблица 5-20 Тактовая память

Номер бита	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Имя переменной								
Период (с)	2.0	1.6	1.0	0.8	0.5	0.4	0.2	0.1
Частота (Гц)	0.5	0.625	1	1.25	2	2.5	5	10

Поскольку тактовая память работает асинхронно к циклу CPU, состояние тактовой памяти может изменяться несколько раз в течение длительного цикла.

5.1.6 Буфер диагностики

CPU поддерживает буфер диагностики, который содержит запись для каждого диагностического события. Каждая запись содержит дату и время, в которое произошло событие, категорию события и его описание. Записи отображаются в хронологическом порядке, причем самое последнее событие находится на самом верху. В этом буфере хранится до 50 актуальных событий. Когда буфер заполняется, новое событие заменяет в нем самое старое. При отключении питания события сохраняются.

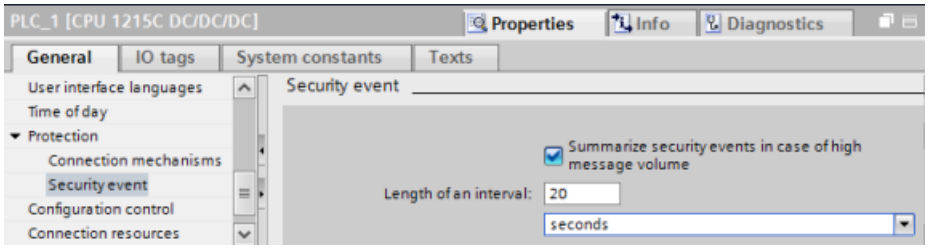
В диагностический буфер записываются следующие типы событий:

- Каждое диагностическое событие в системе, ошибки CPU и модулей
- Каждое изменение состояния CPU (запуск, переход в рабочее состояние STOP или RUN)

Для доступа к диагностическому буферу (Страница 1397) необходимо находиться в режиме онлайн. Буфер находится здесь: "Онлайн и диагностика > Диагностика > Буфер диагностики".

Сокращение количества диагностических событий безопасности

Некоторые события безопасности дублируют записи в буфере диагностики. Такие сообщения могут заполнить буфер диагностики и препятствовать показу других сообщений. Можно настроить PLC для ограничения количества диагностических сообщений из-за событий безопасности. Выбор в конфигурации устройства CPU делается на основе временного интервала, в течение которого необходимо блокировать повторяющиеся сообщения:



Если выбрать суммирование событий безопасности за интервал времени, то можно ввести интервал времени в секундах, минутах или часах и числовое значение в диапазоне от 1 до 255.

Если выбрать ограничение событий безопасности, то ограничению будут подвергнуты следующие типы событий:

- Выход в онлайн с правильным или неправильным паролем
- Обнаружение измененных коммуникационных данных
- Обнаружение измененных данных на карте памяти
- Обнаружение изменений в файле обновления прошивки
- Изменение загруженной в CPU степени защиты (защита доступа)
- Легитимация пароля ограничена или активирована (по инструкции или через дисплей CPU)
- Онлайн-доступ запрещен из-за превышения разрешенного количества одновременных попыток доступа
- Тайм-аут, когда существующее онлайн-соединение не активно
- Вход на веб-сервер с правильным или неправильным паролем
- Создание резервной копии CPU
- Восстановление конфигурации CPU

5.1.7 Часы реального времени

Модуль CPU оснащен часами реального времени. Когда CPU выключен, суперконденсатор поставляет энергию, необходимую для поддержания работы часов. При включенном CPU суперконденсатор заряжается. После того, как CPU был включен по крайней мере на 24 часа, суперконденсатор имеет достаточный заряд, чтобы поддерживать работу часов в течение, как правило, 20 дней.

STEP 7 устанавливает для часов реального времени системное время, которое имеет значение по умолчанию, установленное на заводе, или значение по умолчанию после сброса на заводские настройки. Для использования часов реального времени, сначала необходимо их настроить. Отметки времени, например для записей буфера диагностики, файлов журнала данных и записей журнала данных, основаны на системном времени. Для установки часов реального времени используется функция "Установка времени" (Страница 1393) в окне "Онлайн и диагностика" интерактивного CPU. После этого STEP 7 вычисляет системное время на основе установленного времени плюс-минус часовой пояс операционной системы Windows по UTC (Coordinated Universal Time). Настройка часов реального времени на текущее местное время переводит системное время на UTC, если параметры настройки ОС Windows для часового пояса и летнего времени соответствуют месту установки.

В STEP 7 есть инструкции (Страница 346) для чтения и записи системного времени (RD_SYS_T и WR_SYS_T), для чтения местного времени (RD_LOC_T) и для установки часового пояса (SET_TIMEZONE). Инструкция RD_LOC_T вычисляет местное время на основе часового пояса и временного сдвига для летнего/зимнего времени, введенных в конфигурации часов реального времени в Общих свойства CPU (Страница 167). Эти настройки позволяют установить часовой пояс для местного времени, дополнительно включить переход на летнее время, а также указать даты начала и окончания перехода на летнее время. Также можно использовать инструкции SET_TIMEZONE для настройки этих параметров.

5.1.8 Конфигурирование выходов для перехода из RUN в STOP

Можно сконфигурировать поведение цифровых и аналоговых выходов, когда CPU находится в рабочем состоянии STOP. Для любого выхода CPU, SB или SM можно выбрать деактивацию значений выходов или использование замещающего значения:

- Подстановка замещающего выходного значения (по умолчанию): Ввод замещающего значения для каждого выхода (канала) CPU, SB или SM. Замещающее значение по умолчанию для цифровых выходов ВЫКЛ, для аналоговых выходов равно 0.
- Деактивация выходов в последнем состоянии: Выходы сохраняют свое текущее значение на момент перехода из RUN в STOP. После запуска выходы устанавливаются на замещающее значение по умолчанию.

Настройка поведения выходов выполняется в конфигурации устройства. Выбрать отдельные устройства и открыть вкладку "Свойства", чтобы сконфигурировать выходы для каждого устройства.

Примечание

Некоторые модули распределенного ввода/вывода предлагают дополнительные настройки для реакции на остановку CPU. Выбрать для этих модулей соответствующую опцию из списка в конфигурации устройства.

При переходе CPU из RUN в STOP, он сохраняет образ процесса и записывает соответствующие значения для цифровых и для аналоговых выходов согласно конфигурации.

5.2 Хранение данных, области памяти, I/O и адресация

5.2.1 Доступ к данным S7-1200

STEP 7 упрощает символическое программирование. Для этого создаются символьные имена или "переменные" ("теги") для адресов данных, либо в форме переменных (тегов) PLC, относящихся к адресам памяти и точкам ввода/вывода, либо как локальные переменные, используемые в блоке кода. Для использования этих тегов в программе пользователя необходимо просто ввести имя тега для требуемого параметра инструкции.

Для лучшего понимания того, как CPU структурирует и обращается к областям памяти, ниже объясняется, как теги PLC ссылаются на "абсолютную" адресацию. CPU предлагает несколько вариантов хранения данных во время выполнения программы пользователя:

- Глобальная память: CPU предлагает множество специализированных областей памяти, включая входы (I), выходы (Q) и битовую память (меркеры, M). Эта память доступна для всех блоков кода без ограничений.
- Таблица переменных (тегов) PLC В таблице переменных (тегов) PLC в STEP 7 для определенных адресов в памяти можно вводить символьные имена. Эти переменные (теги) являются глобальными для всей программы STEP 7 и позволяют выполнять программирование с информативными именами.
- Блок данных (DB): В программу пользователя можно вставлять DB для сохранения данных для блоков кода. Сохраненные данные сохраняются после выполнения соответствующего блока кода. В "глобальном" DB хранятся данные, которые могут использоваться всеми блоками кода, а в DB экземпляра хранятся только данные для конкретного FB и он структурируется согласно параметрам FB.
- Временная память: При каждом вызове блока кода, операционная система CPU выделяет временную или локальную память (L), которая будет использоваться при обработке блока. Когда выполнение блока кода заканчивается, CPU перераспределяет локальную память для выполнения других блоков кода.

Каждая отдельная ячейка памяти имеет уникальный адрес. Программа пользователя использует эти адреса для доступа к информации в ячейке памяти. Через ссылки на области памяти для входов (I) или выходов (Q), напр., I0.3 или Q1.7, происходит обращение к образу процесса. Для прямого доступа к физическому входу или выходу добавить ссылку ":P" (например, I0.3:P, Q1.7:P или "Stop:P").

Таблица 5-21 Области памяти

Область памяти	Описание	Принудительное присваивание значений	С сохранением
I Образ процесса входов	Копируется в начале цикла из физических входов	Нет	Нет
I_:P [1] (физический вход)	Прямое чтение физических входов CPU, SB или SM	Да	Нет

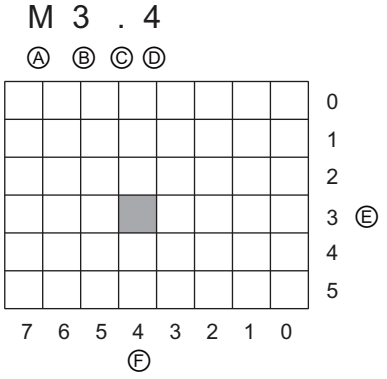
Область памяти	Описание	Принудительное присваивание значений	С сохранением
Q Образ процесса выходов Q_:P [1] (физический выход)	Копируется в начале цикла на физические выходы	Нет	Нет
	Прямая запись на физические выходы CPU, SB или SM	Да	Нет
M Битовая память (мерке-ры)	Управление и хранение данных	Нет	Да (опция)
L Временная память	Временные локальные данные для блока	Нет	Нет
DB Блок данных	Память данных и параметров для FB	Нет	Да (опция)

[1] Для прямого доступа (по чтению или записи) к физическим входам и выходам, добавить код ":P" к адресу или переменной (тегу) (например, I0.3:P, Q1.7:P или "Stop:P").

Каждая отдельная ячейка памяти имеет уникальный адрес. Программа пользователя использует эти адреса для доступа к информации в ячейке памяти. Абсолютный адрес состоит из следующих элементов:

- Идентификатор области памяти (такой как, I, Q или M)
- Размер данных, к которым выполняется обращение (напр., "B" для Byte, "W" для Word или "D" дляDWord)
- Начальный адрес данных (например, байт 3 или слово 3)

При обращении к биту в адресе для логического значения не вводится мнемоника для размера. Вводится только область памяти, местоположение байта и местоположение бита для данных (например, I0.0, Q0.1 или M3.4).



A	Идентификатор области памяти	E	Байты области памяти
B	Адрес байта: Байт 3	F	Биты выбранного байта
C	Разделительный символ ("Байт.Бит")		
D	Местоположение бита в байте (бит 4 из 8)		

В этом примере за область памяти и адресом байта (M = битовая область и 3 = Byte 3) следует точка ("."), отделяющая адрес бита (бит 4).

Обращение к данным в областях памяти CPU

STEP 7 упрощает символическое программирование. Обычно переменные (теги) создаются в таблице переменных (тегов) PLC, блоке данных или на интерфейсе OB, FC или FB. Эти переменные (теги) включают в себя имя, тип данных, смещение и комментарий. Кроме того, в блоке данных может быть указано стартовое значение. Можно использовать эти переменные (теги) при программировании, вводя имя переменной (тега) как параметр для инструкции. Дополнительно можно указать абсолютный операнд (область памяти, размер и смещение) как параметр инструкции. Примеры в следующих разделах показывают, как вводятся абсолютные операнды. Символ % вставляется редактором программы автоматически перед абсолютным операндом. В редакторе программы можно выбрать одно из представлений: символьное, символьное и абсолютное или абсолютное.

I (образ процесса входов): CPU опрашивает периферийные входы (физические входы) непосредственно перед выполнением OB программного цикла в каждом цикле и записывает эти значения в образ процесса входов. К образу процесса входов можно обращаться в формате бита, байта, слова и двойного слова. Возможно обращение как по чтению, так и по записи, но для входов образа процесса как правило используется только чтение.

Таблица 5-22 Абсолютная адресация для областей памяти I

Бит	I[адрес байта].[адрес бита]	I0.1
Байт, слово или двойное слово	I[размер][адрес начального байта]	IB4, IW5 или ID12

Добавляя к адресу ":P", можно напрямую считывать цифровые и аналоговые входы CPU, SB, SM или распределенного модуля. Доступ через I_:P отличается от доступа через I тем, что данные поступают непосредственно с входов, к которым производится обращение, а не из образа процесса входов. Такой доступ через I_:P также называется прямым доступом по чтению, так как данные считываются прямо из источника, а не из его копии, которая была сделана при последнем обновлении образа процесса входов.

Так как физические входы получают свои значения непосредственно из подключенных к ним полевых устройств, то запись на эти входы запрещена. То есть доступ через I_:P является доступом только по чтению, в отличие от доступа к входам, который возможен как по чтению, так и по записи.

Доступ через I_:P ограничен также размером входов, поддерживаемых CPU, SB или SM (с округлением до следующего байта). Например, если входы SB с 2 DI / 2 DQ сконфигурированы так, что они начинаются с I4.0, то обратиться к этим входам можно с помощью I4.0:P и I4.1:P или IB4:P. Обращение к I4.2:P ... I4.7:P не распознается как ошибка, но не имеет смысла, так как эти адреса не используются. Попытки обращения к IW4:P и ID4:P запрещены, так как они превышают байтовое смещение этой SB.

Обращение через I_:P не влияет на соответствующее значение, хранящееся в образе процесса входов.

Таблица 5-23 Абсолютная адресация для областей памяти I (прямая)

Бит	I[адрес байта].[адрес бита]:P	I0.1:P
Байт, слово или двойное слово	I[размер][адрес начального байта]:P	IB4:P, IW5:P, или ID12:P

Q (образ процесса выходов): CPU копирует сохраненные в образе процесса выходов значения на физические выходы. К образу процесс выходов можно обращаться в формате бита, байта, слова и двойного слова. К выходам образа процесса разрешается доступ как по чтению, так и по записи.

Таблица 5-24 Абсолютная адресация для областей памяти Q

Бит	Q[адрес байта].[адрес бита]	Q1.1
Байт, слово или двойное слово	Q[размер][адрес начального байта]	QB5, QW10, QD40

Добавляя к адресу ":P", можно выполнять запись напрямую на цифровые и аналоговые выходы CPU, SB, SM или распределенного модуля. Доступ через Q_:P отличается от доступа через Q тем, что данные поступают непосредственно на выходы, к которым происходит обращение, и дополнительно в образ процесса выходов (Запись выполняется в обе области). Такой доступ через Q_:P также называют прямым доступом по записи, так как данные передаются напрямую на целевой адрес, которому не приходится ждать следующего обновления образа процесса выходов.

Так как физические выходы напрямую управляют полевыми устройствами, подключенными к этим выходам, то чтение с этих выходов запрещено. То есть доступ через Q_:P является доступом только по записи, в отличие от доступа к выходам, который возможен как по чтению, так и по записи.

Доступ через Q_:P ограничен также размером выходов, поддерживаемых CPU, SB или SM (с округлением до следующего байта). Например, если выходы SB с 2 DI / 2 DQ сконфигурированы так, что они начинаются с Q4.0, то обратиться к этим выходам можно с помощью Q4.0:P и Q4.1:P или QB4:P. Обращение к Q4.2:P ... Q4.7:P не распознается как ошибка, но не имеет смысла, так как эти адреса не используются. Попытки обращения к QW4:P и QD4:P запрещены, так как они превышают байтовое смещение этой SB.

Доступ через Q_:P влияет как на физический выход, так и на соответствующее значение, сохраненное в образе процесса выходов.

Таблица 5-25 Абсолютная адресация для областей памяти Q (прямая)

Бит	Q[адрес байта].[адрес бита]:P	Q1.1:P
Байт, слово или двойное слово	Q[размер][адрес начального байта]:P	QB5:P, QW10:P или QD40:P

M (область битовой памяти (M, меркеров): Область памяти меркеров (M) можно использовать для управляющих реле и данных, чтобы хранить промежуточные результаты операций или другую управляющую информацию. К области битовой памяти

можно обращаться в формате бита, байта, слова и двойного слова. К битовой M памяти разрешается доступ как по чтению, так и по записи.

Таблица 5-26 Абсолютная адресация для области памяти M

Бит	M[адрес байта].[адрес бита]	M26.7
Байт, слово или двойное слово	M[размер][адрес начального байта]	MB20, MW30, MD50

Temp (временная память): CPU выделяет временную память по мере необходимости. CPU выделяет временную память блоку кода и инициализирует ячейки памяти в 0 в то время, когда он запускает блок кода (для OB) или вызывает блок кода (для FC или FB).

Временная память подобна битовой M памяти. Но есть важное отличие: M память является "глобальной" , а временная память - "локальной":

- M память: Любой OB, FB и любая FC может обратиться к данным в M памяти, т.е. данные доступны глобально всем элементам программы пользователя.
- Временная память: CPU ограничивает доступ к данным временной памяти теми OB, FB или той FC, где были созданы или объявлены адреса во временной памяти. Адреса во временной памяти остаются локальными и не могут совместно использоваться различными блоками кода, даже если один блок кода вызывает другой блок кода. Пример: Если OB вызывает FC, то FC не может обратиться к временной памяти OB, из которого она вызвана.

CPU предоставляет временную (локальную) память для каждого из классов приоритетов OB:

- 16 кбайт для запуска и программного цикла, включая соответствующие FB и FC
- 6 кбайт для каждого дополнительного потока события прерывания, включая соответствующие FB и FC

Обращение к временной памяти возможно только с использованием символьной адресации.

Через структуру вызова в STEP 7 можно определить объем временной памяти (локальных данных), занимаемой блоками в программе. Выбрать в дереве проекта "Информацию о программе" и после вкладку "Структура вызова". Там отображаются все OB программы и можно показать те блоки, которые вызываются из OB. Для каждого блока отображаются согласованные локальные данные. Для отображения структуры вызова можно использовать и STEP 7 с командой меню **Сервис > Структура вызова**.

DB (блок данных): Блоки данных используются для хранения различных типов данных, включая промежуточные результаты операций или другие управляющие параметры для FB, и структур данных, необходимых для многих инструкций, например, таймеров и счетчиков. К блокам данных можно обращаться в формате бита, байта, слова и двойного слова. Для не защищенных от записи блоков данных разрешается доступ как по чтению,

так и по записи. Доступ к блокам данных, защищенным от записи, возможен только по чтению.

Таблица 5-27 Абсолютная адресация для области памяти DB

Бит	DB[номер блока данных].DBX[адрес байта].[адрес бита]	DB1.DBX2.3
Байт, слово или двойное слово	DB[номер блока данных].DB[размер][адрес начального байта]	DB1.DBB4, DB10.DBW2, DB20.DBD8

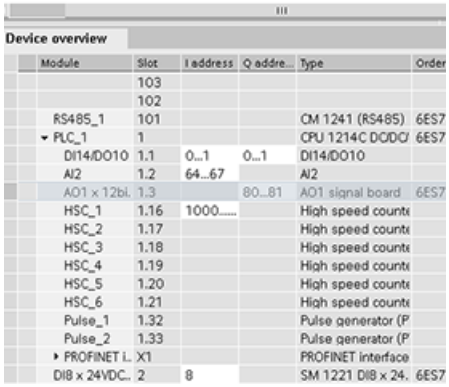
Примечание

Когда абсолютный адрес указывается в LAD или FBD, STEP 7 добавляет перед этим адресом символ "%", чтобы показать, что это - абсолютный адрес. При программировании можно ввести абсолютный адрес с символом "%"или без него (например, %I0.0 или I.0). Если он пропущен, STEP 7 вставляет символ "%".

В SCL необходимо ввести "%" перед адресом, чтобы указать, что это - абсолютный адрес. Без "%" STEP 7 генерирует ошибку неопределенной переменной (тега) во время компиляции.

Конфигурирование I/O в CPU и в модулях ввода-вывода





При добавлении CPU и модулей ввода-вывода в конфигурацию устройства, STEP 7 автоматически назначает адреса I и Q. Можно изменить значение адреса по умолчанию, выбрав поле адреса в конфигурации устройства и введя новые значения.

- STEP 7 распределяет цифровые входы и выходы по группами из 8 каналов (1 байт), независимо от того использует ли модуль все каналы (входы / выходы) или нет.
- STEP 7 назначает аналоговые входы и выходы группами по 2, где каждый аналоговый канал занимает 2 байта (16 бит).

На рисунке показан пример для CPU 1214C с двумя SM и одной SB. В этом примере можно было бы изменить адрес модуля DI8 на 2 вместо 8. Инструмент помогает пользователю, меняя диапазоны адресов, которые являются неподходящими по размеру или конфликтуют с другими адресами.

5.3 Обработка аналоговых значений

Модули аналоговых сигналов подают входные сигналы или ожидают выходные значения, которые представляют собой диапазон значений напряжения или тока. Используются следующие диапазоны: ±10 В, ±5 В, ±2,5 В или от 0 до 20 мА. Значения, возвращенные модулями, являются целыми числами. При этом значения от 0 до 27648 - это номинальный диапазон для тока, а значения от -27648 до 27648 - для напряжения. Значения, находящиеся вне этих диапазонов, представляют собой выход сигнала за верхний или нижний установленный предел. Для получения дополнительной информации о типах значений, находящихся вне диапазона, см. таблицы Представление аналогового входа (Страница 1548) и Представление аналогового выхода (Страница 1549).

В управляющей программе может потребоваться использование этих значений в физических единицах, например чтобы представить объем, температуру, вес или другое количественное значение. Для этого сначала необходимо нормализовать аналоговый вход в действительное число (значение с плавающей запятой) между 0,0 и 1,0. После следует масштабировать значение для минимального и максимального значения представляемой физической единицы. Для значений в физических единицах, которые должны быть преобразованы в аналоговое выходное значение, сначала следует нормализовать значение в физических единицах в значение между 0,0 и 1,0. После выполняется масштабирование значения между 0 и 27648 или -27648 и 27648, в зависимости от диапазонов аналогового модуля. В STEP 7 для этого есть инструкции NORM_X и SCALE_X (Страница 305). Можно использовать и инструкцию CALCULATE (Страница 247) для масштабирования аналоговых значений (Страница 41).

Пример: Обработка аналоговых значений

Рассмотрим в качестве примера аналоговый вход с диапазоном тока 0 - 20 мА. Модуль аналогового ввода возвращает измеренные значения в диапазоне от 0 до 27648. В этом примере мы исходим из того, что входное аналоговое значение будет использоваться для измерения температурного диапазона от 50 °C до 100 °C. Несколько выборочных значений будут иметь следующие соответствия:

Аналоговое входное значение	Технические единицы
0	50 °C
6192	62,5 °C
12384	75 °C
18576	87,5 °C
27648	100 °C

Физические единицы в этом примере были вычислены на основе входного аналогового значения следующим образом:

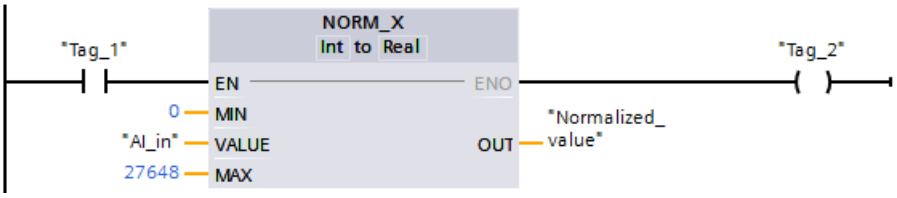
Значение в физических единицах = 50 + (входное аналоговое значение) * (100 - 50) / (27648 - 0)

В общей форме уравнение будет выглядеть следующим образом:

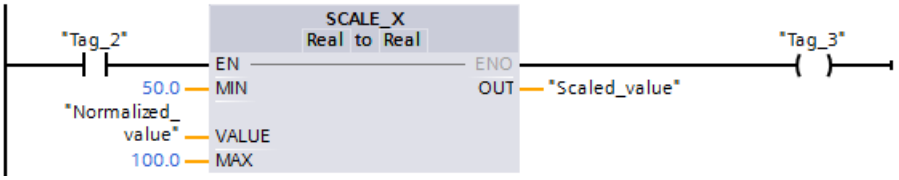
Значение в физических единицах = (нижний диапазон физических единиц) +
(аналоговое входное значение) *
(верхний диапазон физических единиц - нижний диапазон физических единиц) /
(максимальный аналоговый входной диапазон - минимальный аналоговый входной диапазон)

В приложениях PLC входное аналоговое значение обычно нормализуется в значение с плавающей запятой между 0,0 и 1,0. После результат преобразуется в значение с плавающей запятой в диапазоне физических единиц. Для упрощения следующие LAD операторы используют постоянные значения для диапазонов; но могут использоваться и переменные (теги).

Сегмент 1



Сегмент 2



5.4 Типы данных

Типы данных используются для указания размера элемента данных, а также для того, как эти данные могут быть интерпретированы. Каждый параметр инструкции поддерживает, по крайней мере, один тип данных, а некоторые параметры поддерживают несколько типов данных. Установить указатель мыши на поле параметров инструкции, чтобы увидеть, какие типы данных поддерживаются для соответствующего параметра.

Формальный параметр – это идентификатор на инструкции, который указывает адрес данных, подлежащих использованию инструкцией (пример: вход IN1 инструкции ADD). Фактический параметр – это адрес (начинающиеся с символа "%") или константа, где содержатся данные, подлежащие использованию инструкцией (пример: %MD400 "Number_of_Widgets"). Тип данных указанного фактического параметра должен соответствовать одному из поддерживаемых типов данных формального параметра, определяемого инструкций.

При указании фактического параметра указывается либо переменная (символ), либо абсолютный (прямой) адрес. Переменные (теги) связывают символьное имя (имя переменной (тега)) с типом данных, областью памяти, смещением в памяти и

комментарием и могут быть созданы в редакторе переменных (тегов) PLC или в редакторе интерфейса блока (OB, FC, FB и DB). Если вводится абсолютный адрес, не связанный ни с какой переменной (тегом), необходимо использовать подходящий размер, соответствующий поддерживаемому типу данных, тогда при вводе создается стандартная переменная (тег).

Все типы данных, за исключением String, Struct, Array иDTL, доступны в редакторе переменных (тегов) PLC и на интерфейсах блоков. String, Struct, Array и DTL доступны только на интерфейсах блоков. Для многих входных параметров можно также вводить постоянные значения.

- Бит и комбинации битов (Страница 125): Bool (логическое значение или битовое значение), Byte (цепочка из 8 бит), Word (цепочка из 16 бит), DWord (цепочка из 32 бит)
- Целое число (Страница 126)
  - USInt (8-битовое целое число без знака), SInt (8-битовое целое число со знаком),
  - UInt (16-битовое целое число без знака), Int (16-битовое целое число со знаком),
  - UDInt (32-битовое целое число без знака), DInt (32-битовое целое число со знаком),
- Число с плавающей запятой/действительное число (Страница 126): Real (32-битовое действительное значение или значение с плавающей точкой), LReal (64-битовое действительное значение или значение с плавающей точкой)
- Время и дата (Страница 127): Time (32-битовое значение времени по МЭК), Date (16-битовое значение даты), TOD (32-битовое значение времени), DTL (12-байтовая структура даты и времени)
- Символ и строка (Страница 129): Char (8-битовый одиночный символ), String (строка переменной длины до 254 символов)
- Массив (Страница 131)
- Структура данных (Страница 132): Struct
- Тип данных PLC (Страница 132)
- Тип данных Variant (Страница 133)

Следующий числовой формат BCD поддерживается инструкциями преобразования, хотя он и не доступен как тип данных.

Таблица 5-28 Размер и диапазон формата BCD

Формат	Размер (в битах)	Диапазон	Примеры ввода констант
BCD16	16	От -999 до 999	123, -123
BCD32	32	От -9999999 до 9999999	1234567, -1234567

5.4.1 Типы данных Bool, Byte, Word и DWord

Таблица 5-29 Битовые типы данных

Тип данных	Размер в битах	Тип числа	Диапазон числа	Примеры для констант	Примеры для адресов
Bool	1	Булевый	ЛОЖИ или ИСТИНА	ИСТИНА	I1.0 Q0.1 M50.7 DB1.DBX2.3 имя переменной (тега)
		Двоичный	2#0 или 2#1	2#0	
		Целое число без знака	0 или 1	1	
		Восьмеричный	8#0 или 8#1	8#1	
		Шестнадцатеричный	16#0 или 16#1	16#1	
Byte	8	Двоичный	От 2#0 до 2#1111_1111	2#1000_1001	IB2 MB10 DB1.DBB4 имя переменной (тега)
		Целое число без знака	От 0 до 255	15	
		Целое число со знаком	От -128 до 127	-63	
		Восьмеричный	от 8#0 до 8#377	8#17	
		Шестнадцатеричный	От B#16#0 до B#16#FF, от 16#0 до 16#FF	B#16#F, 16#F	
Word	16	Двоичный	От 2#0 до 2#1111_1111_1111_1111	2#1101_0010_1001_0110	MW10 DB1.DBW2 имя переменной (тега)
		Целое число без знака	От 0 до 65535	61680	
		Целое число со знаком	От -32768 до 32767	72	
		Восьмеричный	От 8#0 до 8#177_777	8#170_362	
		Шестнадцатеричный	От W#16#0 до W#16#FFFF, от 16#0 до 16#FFFF	W#16#F1C0, 16#A67B	
DWord	32	Двоичный	От 2#0 до 2#1111_1111_1111_1111_1111_1111_1111_1111	2#1101_0100_1111_1110_1000_1100	MD10 DB1.DBD8 имя переменной (тега)
		Целое число без знака*	От 0 до 4_294_967_295	15_793_935	
		Целое число со знаком*	От -2_147_483_648 до 2_147_483_647	-400000	
		Восьмеричный	От 8#0 до 8#37_777_777_777	8#74_177_417	
		Шестнадцатеричный	От DW#16#0000_0000 до DW#16#FFFF_FFFF, от 16#0000_0000 до 16#FFFF_FFFF	DW#16#20_F30A, 16#B_01F6	

* Символ подчеркивания "_" разделяет тысячные разряды для улучшения читабельности чисел с более чем восьмью цифрами.

5.4.2 Целочисленные типы данных

Таблица 5-30 Целочисленные типы данных (U = без знака, S = короткое, D = двойное)

Тип данных	Размер в битах	Диапазон числа	Примеры для констант	Пример для адресов
USInt	8	От 0 до 255	78, 2#01001110	MB0, DB1.DBB4, имя переменной (тега)
SInt	8	От -128 до 127	+50, 16#50	
UInt	16	От 0 до 65.535	65295, 0	MW2, DB1.DBW2, имя переменной (тега)
Int	16	От -32.768 до 32.767	30000, +30000	
UDInt	32	От 0 до 4.294.967.295	4042322160	MD6, DB1.DBD8, имя переменной (тега)
DInt	32	От -2.147.483.648 до 2.147.483.647	-2131754992	

5.4.3 Типы данных для действительных чисел с плавающей запятой

Вещественные числа (или числа с плавающей точкой) представлены как 32-разрядные числа одинарной точности (Real) или как 64-разрядные числа двойной точности (LReal), согласно описанию в стандарте/норме ANSI/IEEE7541985. Числа одинарной точности с плавающей запятой имеют точность до 6 значащих цифр, а числа с плавающей запятой двойной точности имеют точность до 15 значащих цифр. Можете указать максимум 6 (Real) или 15 (LReal) значащих цифр при вводе константы с плавающей запятой.

Таблица 5-31 Типы данных для действительных чисел с плавающей запятой (L=длинное)

Тип данных	Размер в битах	Диапазон числа	Примеры для констант	Примеры для адресов
Real	32	От -3.402823e+38 до -1.175 495e-38, ±0, от +1.175 495e-38 до +3.402823e +38	123.456, -3.4, 1,0e-5	MD100, DB1.DBD8, имя переменной (тега)
LReal	64	От -1.7976931348623158e+308 до -2.2250738585072014e-308, ±0, от +2.2250738585072014e-308 до +1.7976931348623158e+308	12345,123456789e40, 1,2E+40	DB_имя.переменная_имя Правила: • Прямая адресация не поддерживается • Назначение на интерфейсе блока OB, FB или FC

Вычисления, для которых требуется длинная последовательность значений, включая очень большие и очень малые числа, могут привести к неточным результатам. Это может

произойти, если числа отличаются в 10 в степени x раз, где x > 6 (Real) или 15 (LReal). Пример (Real): 100 000 000 + 1 = 100 000 000.

5.4.4 Типы данных времени и даты

Таблица 5-32 Типы данных времени и даты

Тип данных	Размер	Диапазон	Примеры ввода констант
Time	32 бита	От T#-24d_20h_31m_23s_648ms до T#24d_20h_31m_23s_647ms Сохраняется как: от -2.147.483.648 мс до +2.147.483.47 мс	T#5m_30s T#1d_2h_15m_30s_45ms TIME#10d20h30m20s630ms 500h10000ms 10d20h30m20s630ms
Date	16 битов	От D#1990-1-1 до D#2168-12-31	D#2009-12-31 DATE#2009-12-31 2009-12-31
Time_of_Day	32 бита	От TOD#0:0:0.0 до TOD#23:59:59.999	TOD#10:20:30.400 TIME_OF_DAY#10:20:30.400 23:10:1
DTL (Date and Time Long)	12 байтов	Мин.: DTL#1970-01-01-00:00:00.0 Макс.: DTL#2262-04-11:23:47:16.854 775 807	DTL#2008-12-16-20:30:20.250

Время

Данные TIME хранятся как двойное целое число со знаком, интерпретируемое как миллисекунды. Формат редактора может использовать информацию о дне (d), часах (h), минутах (m), секундах (s) и миллисекундах (ms).

Указание всех единиц времени не обязательно. Например, допустимы T#5h10s и 500h.

Общее из всех указанных значений в отдельных единицах не должно выходит за границы верхнего или нижнего предельного значения в миллисекундах для типа данных времени Time (от -2.147.483.648 мс до +2.147.483.647 мс).

Дата

Данные DATE хранятся как целое значение без знака, которое интерпретируется как число дней, добавленных к исходной дате 01.01.1990 , чтобы получить заданную дату. Формат редактора должен определить год, месяц и день.

TOD

Данные TOD (TIME_OF_DAY) данные хранятся как двойное целое число без знака, которое интерпретируется как число миллисекунд с полуночи до заданного времени суток (полночь = 0 мс). Должны быть указаны часы (24 часа/день), минуты и секунды. Указание долей секунд по желанию.

DTL

Тип данных DTL (Date and Time Long) использует 12-байтовую структуру для сохранения информации о дате и времени. Можно определить тип данных DTL либо во временной памяти блока, либо в DB. Для всех компонентов в столбце для стартового значения в редактор DB должно быть введено значение.

Таблица 5-33 Размер и диапазон DTL

Длина (в байтах)	Формат	Диапазон значений	Пример ввода значения
12	Часы и календарь Год-Месяц-День:Час:Минута:Секунда.Наносекунды	Мин.: DTL#1970-01-01-00:00:00.0 Макс.: DTL#2554-12-31-23:59:59.999 999 999	DTL#2008-12-16-20:30:20.250

Каждый компонент типа данных DTL имеет отличный тип данных и диапазон значений. Тип данных указанного значения должен соответствовать типу данных соответствующего компонента.

Таблица 5-34 Содержание структуры DTL

Байт	Компонент	Тип данных	Диапазон значений
0	Год	UINT	От 1970 до 2554
1			
2	Месяц	USINT	От 1 до 12
3	День	USINT	От 1 до 31
4	День недели[1]	USINT	От 1 (воскресенье) до 7 (суббота)[1]
5	Час	USINT	От 0 до 23
6	Минута	USINT	От 0 до 59
7	Секунда	USINT	От 0 до 59
8	Наносекунды	UDINT	От 0 до 999.999.999
9			
10			
11			

[1] Формат Год-Месяц-День:Час:Минута:Секунда.Наносекунда не включает в себя день недели.

5.4.5 Символьные и строковые типы данных

Таблица 5-35 Символьные и строковые типы данных

Тип данных	Размер	Диапазон	Примеры ввода констант
Char	8 битов	От 16#00 до 16#FF	'A', 't', '@', 'ä', '∑'
WChar	16 битов	От 16#0000 до 16#FFFF	'A', 't', '@', 'ä', '∑', азиатские символы, кириллические символы и пр.
String	n+ 2 байтов	n = (от 0 до 254 байтов)	"ABC"
WString	n+ 2 слов	n = (от 0 до 65534 слов)	"ä123@XYZ.COM"

Char и WChar

Char (символ) занимает один байт в памяти и хранит единственный символ, закодированный в формате ASCII, включая расширенные коды ASCII символов. WChar занимает одно слово в памяти и может содержать любое представление символа в виде двух байтов.

Синтаксис редактора использует символ одинарной кавычки до и после символа. Можно использовать видимые символы и управляющие символы.

String и WString

CPU поддерживает тип данных String для хранения последовательности однобайтовых символов. Тип данных String содержит общее количество символов (число символов в строке) и текущее количество символов. Тип данных String предлагает до 256 байтов для хранения максимального общего количества символов (1 байт), текущего количество символов (1 байт) и до 254 байтов в строке. Каждый байт в типе данных String может быть любым значением от 16#00 до 16#FF.

Тип данных WString предусматривает более длинные строки из значений по одному слову (двойной байт). Первое слово содержит максимальное общее количество символов; следующее слово содержит текущее общее количество символов и следующая строка может содержать до 65334 слов. Каждое слово в типе данных WString может быть любым значением от 16#0000 до 16#FFFF.

Можно указывать буквенные строки (константы) для параметров инструкции типа IN в одинарных кавычках. Например, 'ABC' - это строка из трех символов, которая может использоваться в качестве входа для параметра IN инструкции S_CONV. Также можно создавать строковые переменные, выбрав тип данных "String" или "WString" в редакторах интерфейса блоков для OB, FC, FB и DB. Нельзя создать строку в редакторе переменных (тегов) PLC.

Можно определить максимальный размер строки в байтах (String) или словах (WString), вставив квадратные скобки после ключевого слова "String" или "WString" после выбора соответствующего типа данных в выпадающем списке. Пример: Запись "MyString String [10]" указывает максимальный размер в 10 байтов для строки MyString. Если не добавлять квадратные скобки с максимальным размером, то для типа данных String используется размер 254, а для типа данных WString - 65534. Запись "MyWString WString[1000]" указывает на тип данных WString длиной в 1000 слов.

Следующий пример показывает строку с максимальным количеством символов 10 и текущим количеством символов 3. Это означает, что строка в настоящее время содержит 3 однобайтовых символа, но может быть расширена до 10 однобайтовых символов.

Таблица 5-36 Пример для типа данных String

Общее кол-во символов	Текущее кол-во символов	Символ 1	Символ 2	Символ 3	...	Символ 10
10	3	'C' (16#43)	'A' (16#41)	'T' (16#54)	...	-
Байт 0	Байт 1	Байт 2	Байт 3	Байт 4	...	Байт 11

Следующий пример показывает тип данных WString с максимальным количеством символов 500 и текущим количеством символов 300. Строка в настоящее время содержит 300 символов по одному слову, но может быть расширена до 500 символов по одному слову.

Таблица 5-37 Пример для типа данных WString

Общее кол-во символов	Текущее кол-во символов	Символ 1	Символы от 2 до 299	Символ 300	...	Символ 500
500	300	'ä' (16#0084)	Слова ASCII символов	'M' (16#004D)	...	-
Слово 0	Слово 1	Слово 2	Слова от 3 до 300	Слово 301	...	Слово 501

Управляющие ASCII символы могут использоваться в данных Char, Wchar, String и WString. Следующая таблица показывает примеры синтаксиса управляющих символов.

Таблица 5-38 Допустимые управляющие ASCII символы

Управляющие символы	ASCII шестн. знач. (Char)	ASCII шестн. знач. (WChar)	Управляющая функция	Примеры
$L или $l	16#0A	16#000A	Перевод строки	'$LText', '$0AText'
$N или $n	16#0A и 16#0D	16#000A и 16#000D	Разрыв строки Новая строка показывает два символа в строке.	'$NText', '$0A $0DText'
$P или $p	16#0C	16#000C	Подача страницы	'$PText', '$0CText'
$R или $r	16#0D	16#000D	Возврат каретки (CR)	'$RText','$0DText'
$T или $t	16#09	16#0009	Табулятор	'$TText', '$09Text'
$$	16#24	16#0024	Символ доллара	'100$$', '100$24'
$'	16#27	16#0027	Одинарная кавычка	'$'Text$'','$27Text $27'

5.4.6 Тип данных ARRAY

Массивы

Можно создать массив, который содержит несколько элементов одного и того же типа данных. Массивы могут быть созданы на интерфейсе блоков для OB, FC, FB и DB. Нельзя создать массив в редакторе переменных (тегов) PLC.

Для создания массива на интерфейсе блока необходимо присвоить массиву имя и выбрать тип данных "Array [lo .. hi] of type", после изменить "lo", "hi" и "type" следующим образом:

- lo - стартовый (наименьший) индекс для массива
- hi - конечный (наивысший) индекс для массива
- type - один из типов данных, например, BOOL, SINT, UDINT

Таблица 5-39 Правила для типа данных ARRAY

Тип данных	Синтаксис Array		
ARRAY	Имя [индекс1_мин..индекс1_макс, индекс2_мин..индекс2_макс] <тип данных> • Все элементы массива должны иметь один и тот же тип данных. • Индекс может быть отрицательным, но нижнее предельное значение должно быть меньше или равно верхнему предельному значению. • Массивы могут охватывать до шести измерений. • Объявления индексов многомерного массива мин..макс разделяются символами запятой. • Вложенные массивы или массивы из массивов недопустимы. • Объем памяти, занимаемый массивом = (размер одного элемента * общее количество элементов в массиве)		
	Индекс массива	**Допустимые типы данных индекса**	**Правила для индекса массива**
	Константа или переменная	USInt, SInt, UInt, Int, UDInt, DInt	• Предельные значения: от -32768 до +32767 • Допустимо: Комбинированные константы и переменные • Допустимо: Константные выражения • Недопустимо: Переменные выражения

Пример: Объявления массива	ARRAY[1..20] of REAL	Одно измерение, 20 элементов
	ARRAY[-5..5] of INT	Одно измерение, 11 элементов
	ARRAY[1..2, 3..4] of CHAR	Два измерения, 4 элемента
Пример: Адресация элементов массива	ARRAY1[0]	ARRAY1 элемент 0
	ARRAY2[1,2]	ARRAY2 элемент [1,2]
	ARRAY3[i,j]	Если i =3 и j=4, то адресуется элемент ARRAY3[3, 4]

5.4.7 Структурированные типы данных

Можно использовать тип данных "Struct", чтобы определить структуру данных, состоящую из других типов данных. Структурированный тип данных может использоваться, чтобы обработать группу связанных данных процесса как отдельный блок данных. Типу данных Struct присваивается имя и определяется внутренняя структура данных в редакторе блока данных или редакторе интерфейса блока.

Массивы и структуры могут быть собраны в большую структуру. Структура может иметь вложенность до восьми уровней. Например, можно создать структуру из структур, которые в свою очередь содержат массивы.

5.4.8 Тип данных PLC

Редактор типа данных PLC позволяет определять структуры данных, которые можно многократно использовать в программе. Для создания типа данных PLC открыть "Типы данных PLC" в дереве проекта и выбрать пункт "Добавить новый тип данных". Два одиночных клика на вновь создаваемом типе данных PLC, чтобы изменить его имя по умолчанию. Двойной клик, чтобы открыть редактор типов данных PLC.

Для создания пользовательской структуры типа данных PLC можно использовать те же методы, что и в редакторе блока данных. Добавить новые строки для любых типов данных, которые необходимы, чтобы создать нужную структуру данных.

При создании типа данных PLC имя нового типа данных PLC появляется в выпадающих списках выбора в редакторе DB и редакторе интерфейса блока кода.

Возможные варианты использования типов данных PLC:

- Как тип данных на интерфейсе блока кода или в блоках данных
- В качестве шаблона для создания нескольких глобальных блоков данных с идентичной структурой данных
- В качестве типа данных для описаний переменных (тегов) PLC в областях памяти I и Q модуля CPU

Например, тип данных PLC может быть рецептом для смешивания цветов. Можно присвоить этот тип данных PLC нескольким блокам данных. Для создания определенного цвета можно изменять переменные (теги) в каждом блоке данных.

5.4.9 Тип данных указателей "Variant"

Тип данных Variant может ссылать на переменные различных типов данных или параметры. Указатель Variant может указывать на структуры и отдельные компоненты структур. Указатель Variant не занимает мест в памяти.

Таблица 5-40 Свойства указателя Variant

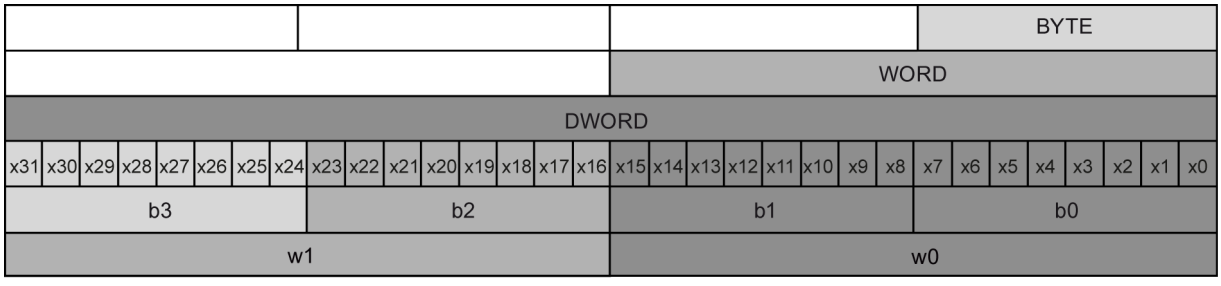
Длина (в байтах)	Представление	Формат	Пример записи
0	Символьное	Операнд	MyTag
		Имя_DB.Имя_структуры.Имя_элемента	MyDB.Struct1.pressure1
	Абсолютное	Операнд	%MW10
		Номер_DB. Операнд Тип Длина	P#DB10.DBX10.0 INT 12

5.4.10 Обращение к "срезу" тегированного типа данных

К переменным (тегам) PLC и переменным (тегам) блока данных можно обращаться на уровне бита, байта или слова в зависимости от их размера. Синтаксис для доступа к такому срезу данных является следующим:

- "<Имя переменной (тега) PLC>".xn (обращение к биту)
- "<Имя переменной (тега) PLC>".bn (обращение к байту)
- "<Имя переменной (тега) PLC>".wn (обращение к слову)
- "Имя блока данных".<Имя переменной (тега)>.xn (обращение к биту)
- "<Имя блока данных>".<Имя переменной (тега)>.bn (обращение к байту)
- "<Имя блока данных>".<Имя переменной (тега)>.wn (обращение к слову)

К переменной (тегу) размера двойного слова данных можно обратиться, используя биты 0 - 31, байты 0 - 3, или слова 0 - 1. К переменной (тегу) размера слова можно обратиться, используя биты 0 - 15, байты 0 - 1 или слово 0. К переменной (тегу) размера байта можно обратиться, используя биты 0 - 7 или байт 0. Битовый, байтовый и пословный срезы могут использоваться везде, где биты, байты или слова являются ожидаемыми операндами.



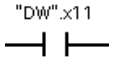
Примечание

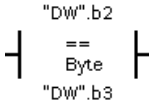
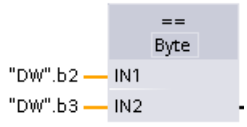
Допустимыми типами данных, к которым можно обратиться посредством среза являются: Byte, Char, Conn_Any, Date, DInt, DWord, Event_Any, Event_Att, Hw_Any, Hw_Device, HW_Interface, Hw_Io, Hw_Pwm, Hw_SubModule, Int, OB_Any, OB_Att, OB_Cyclic, OB_Delay, OB_WHINT, OB_PCYCLE, OB_STARTUP, OB_TIMEERROR, OB_Tod, Port, Rtm, SInt, Time, Time_Of_Day, UDInt, UInt, USInt и Word. К переменным (тегам) PLC с типом данных Real можно обратиться, используя срез, а к переменным (тегам) блока данных с типом Real - нельзя.

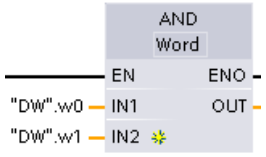
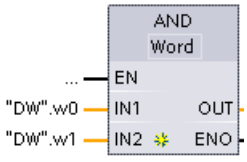
Примеры

В таблице переменных (тегов) PLC, "DW" объявлен как переменная (тег) с типом DWORD. Примеры демонстрируют обращение к биту, байту и слову, используя срез:

	LAD	FBD	SCL

Обращение к биту			`IF "DW".x11 THEN`
			`...`
			`END_IF;`

Обращение к байту			`IF "DW".b2 = "DW".b3`
			`THEN`
			`...`
			`END_IF;`

Обращение к слову			`out:= "DW".w0 AND` `"DW".w1;`

5.4.11 Обращение к переменной (тегу) с использование AT наложения

AT наложение для переменной (тега) позволяет обратиться к уже объявленной переменной (тегу) блока со стандартным доступом, используя накладываемое объявление другого типа данных. Можно, например, адресовать отдельные биты переменной (тега) с типами Byte, Word или DWord с помощью массива из Bool. AT наложения доступны для следующих типов переменных (тегов):

- Переменные (теги) в блоке стандартного доступа
- Сохраняющиеся переменные (теги) в оптимизированном блоке

Объявление

Чтобы выполнить наложение на параметр, следует объявить дополнительный параметр непосредственно после параметра, на который должно быть выполнено наложение и выбрать тип данных "AT" Редактор создает наложение, и после можно выбрать тип данных, структуру или массив для наложения.

Пример

Настоящий пример показывает входные параметры FB со стандартным доступом. На байтовую переменную (тег) B1 выполняется наложение массива булевских элементов:

B1		Byte	0.0
OV	AT"B1"	Array[0..7] of Bool	0.0
OV[0]		Bool	0.0
OV[1]		Bool	0.1
OV[2]		Bool	0.2
OV[3]		Bool	0.3
OV[4]		Bool	0.4
OV[5]		Bool	0.5
OV[6]		Bool	0.6
OV[7]		Bool	0.7

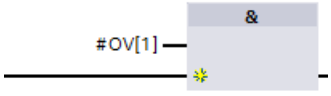
Следующий пример - это переменная (тег) типа DWord с наложением данных типа Struct. Struct содержит одно слово, один байт и два булевых значения:

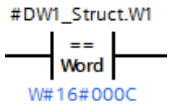
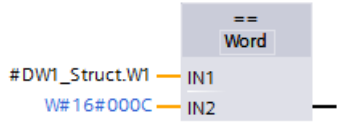
DW1		DWord	2.0
DW1_Struct	AT"DW1"	Struct	2.0
W1		Word	0.0
B1		Byte	2.0
BO1		Bool	3.0
BO2		Bool	3.1

Столбец Offset интерфейса блока показывает расположение наложенных типов данных относительно исходной переменной (тега).

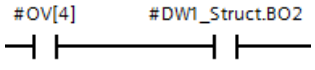
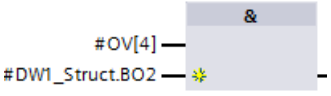
Можно обращаться к наложенным типам напрямую в логике программы:

LAD	FBD	SCL

		`IF #OV[1] THEN`
		`...`
		`END_IF;`

		`IF #DW1_Struct.W1 = W#16#000C THEN`
		`...`
		`END_IF;`

		`out1 := #DW1_Struct.B1;`

		`IF #OV[4] AND #DW1_Struct.BO2 THEN`
		`...`
		`END_IF;`

Правила

- Наложение переменных (тегов) возможно в блоках FB и FC со стандартным (не оптимизированным) доступом.
- В оптимизированных блоках FB и FC возможно наложение сохраняющихся переменных (тегов).
- Возможно наложение параметров для всех типов блоков и всех разделов объявления.
- Можно использовать наложенный параметр как любой другой параметр блока.
- Наложение параметров типа VARIANT невозможно.
- Размер накладываемого параметра должен быть меньше или равен размеру параметра, на который выполняется наложение.
- Необходимо объявить накладываемую переменную сразу после переменной, на которую она накладывается, и выбрать ключевое слово "AT" в качестве исходного типа данных.

5.5 Использование карты памяти

Примечание

Модули CPU поддерживают только предварительно отформатированные карты памяти SIMATIC (Страница 1624).

Перед копированием программ на отформатированную карту памяти следует удалить с карты все ранее сохраненные на ней программы.

Можно использовать карту памяти в качестве трансферной карты или программной карты. Трансферные и программные карты содержит все блоки кода и блоки данных, все технологические объекты и конфигурацию устройств. На трансферных и программных картах **отсутствуют**, напр., таблица принудительно задаваемых значений, таблицы текущего контроля или таблицы переменных (тегов) PLC.

- С помощью трансферной карты (Страница 140) программа копируется во внутреннюю загрузочную память CPU без использования STEP 7.
  С помощью пустой трансферной карты можно обращаться к защищенному паролем CPU, если пароль был утерян или забыт (Страница 149).
- Программная карта (Страница 142) используется как внешняя загружаемая память для CPU.

Карта памяти также используется при обновлении прошивки (Страница 147).

5.5.1 Вставка карты памяти в CPU

ВНИМАНИЕ

Защита карты памяти и слота от электростатического разряда

Электростатические разряды могут повредить карту памяти или предусмотренный для нее слот в CPU.

При работе с картой памяти следует стоять на электропроводящей, заземленной поверхности и/или носить заземленный браслет. Карта памяти должна храниться в токопроводящем контейнере.



Убедиться, что карта памяти не защищена от записи. Для этого сдвинуть переключатель блокировки в сторону от позиции "Lock".

Если вставить защищенную от записи карту памяти в CPU, то STEP 7 при следующем запуске выведет диагностическое сообщение с указанием на это состояние. Модуль CPU выполнит запуск в штатном режиме, но инструкции с рецептами или архивами данных вызывают ошибку при использовании карты с защитой от записи.

⚠ ПРЕДУПРЕЖДЕНИЕ

Перед вставкой карты памяти следует убедиться, что CPU не управляет процессом.

Вставка карты памяти (независимо от того, используется ли она как программная карта, трансферная карта или карта для обновления прошивки) в работающий CPU приводит к немедленному переходу CPU в режим STOP, что может вызвать прерывание процесса и стать причиной смерти или тяжких телесных повреждений.

Прежде, чем вставить или извлечь карту памяти, следует убедиться, что CPU не находится в фазе активного управления машиной или процессом. Всегда следует оборудовать цепь АВАРИЙНОГО ОТКЛЮЧЕНИЯ для приложения или процесса.

Примечание

Не вставлять трансферную карту с программой V3.0 в S7-1200 CPU версии V4.x.

Трансферные карты с программой версии 3.0 несовместимы с S7-1200 CPU версии 4.x. Вставка карты памяти, которая содержит программу V3.0, вызывает ошибку CPU.

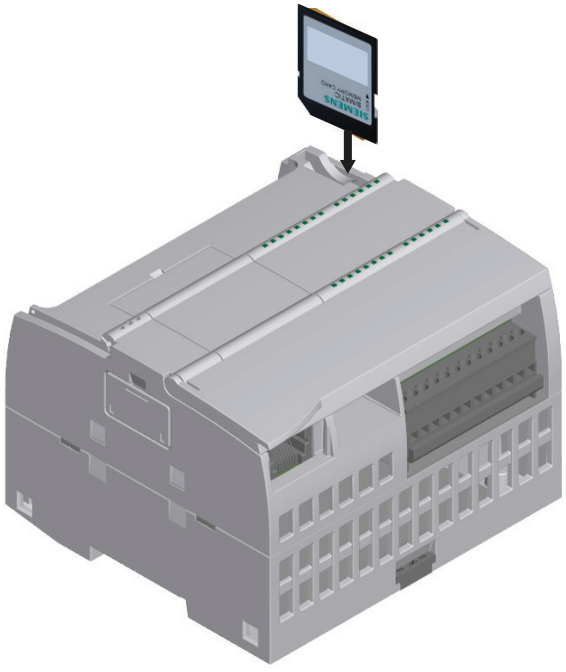
Если была вставлена трансферная карта (Страница 140) с неправильной версией программы, то следует извлечь карту, перевести CPU из STOP в RUN, выполнить восстановление исходного состояния памяти (MRES) или выключить и снова включить CPU. После того, как CPU будет выведен из состояния ошибки, можно загрузить правильную программу CPU V4.x.

Для перевода программы V3.0 в программу V4.x следует использовать TIA Portal, чтобы сменить устройство в аппаратной конфигурации.

Примечание

При вставке карты памяти в CPU, находящемся в режиме STOP, буфер диагностики отображает сообщение о том, что был инициирован анализ карты памяти. CPU выполнит анализ карты памяти при следующем переводе CPU в рабочее состояние RUN, выполнении восстановления исходного состояния памяти CPU через MRES или выключении и повторном включении CPU.

Таблица 5-41 Вставка карты памяти

	Чтобы вставить карту памяти, открыть верхнюю крышку CPU и вставить карту памяти в слот. Нажимной соединитель обеспечивает простую вставку и извлечение карты памяти. Форма карты памяти такова, что она может быть вставлена только надлежащим образом.

Поведение CPU при вставке карты памяти

При вставке карты памяти, CPU выполняет следующие операции:

1. Переходит в рабочее состояние STOP (если уже не находится в состоянии STOP).
2. Появляется запрос на одно из следующих действий:
   - Выключение и повторное включение
   - Переход в рабочее состояние RUN
   - Выполнение восстановления исходного состояния памяти
3. Анализ карты

Анализ карты памяти со стороны CPU

Если в Параметрах защиты конфигурации устройства (Страница 206) модуля CPU не выбрана установка "Отключить копирование из внутренней загружаемой памяти во внешнюю загружаемую память", то CPU определяет, какой тип карты памяти вставляется:

- **Пустая карта памяти**: На пустой карте памяти нет файла заданий (S7_JOB.S7S). При вставке пустой карты памяти, CPU добавляет программный файл заданий. После он копирует внутреннюю загружаемую память во внешнюю загружаемую память (файл программы на карте памяти) и стирает внутреннюю загружаемую память.
- **Пустая программная карта**: Пустая программная карта содержит пустой программный файл заданий. В этом случае CPU копирует внутреннюю загружаемую память во внешнюю загружаемую память (файл программы на карте памяти) и стирает внутреннюю загружаемую память.

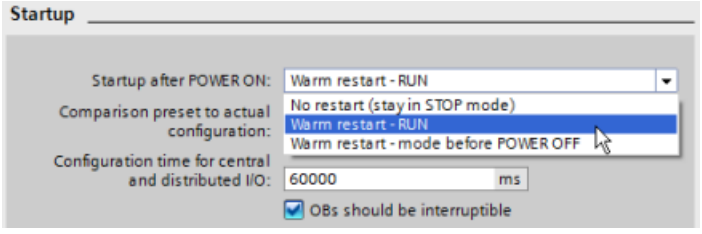
Если в параметрах защиты конфигурации устройства CPU выбрана установка "Отключить копирование из внутренней загружаемой памяти во внешнюю загружаемую память", то CPU ведёт себя следующим образом:

- **Пустая карта памяти**: На пустой карте памяти нет файла заданий (S7_JOB.S7S). При вставке пустой карты памяти CPU не выполняет никаких действий. Он не создает программный файл заданий и не копирует внутреннюю загружаемую память во внешнюю загружаемую память (файл программы на карте памяти). Он не стирает внутреннюю загружаемую память.
- **Пустая программная карта**: Пустая программная карта содержит пустой программный файл заданий. Действия со стороны CPU отсутствуют. Онне копирует внутреннюю загружаемую память во внешнюю загружаемую память (файл программы на карте памяти). Он не стирает внутреннюю загружаемую память.

Если в CPU вставляется программная карта (Страница 142), трансферная карта (Страница 140) иликарта с обновлением прошивки (Страница 147), то настройка "Отключить копирование из внутренней загружаемой памяти во внешнюю загружаемую память" не действует для анализа карты памяти со стороны CPU.

5.5.2 Настройка параметров запуска CPU перед копированием проекта на карту памяти

При копировании программы на трансферную или программную карту, эта программа содержит параметры запуска для CPU. Перед копированием программы на карту памяти следует проверить, сконфигурирован ли режим работы CPU после выключения и последующего включения питания. Можно выбрать состояние (STOP, RUN или последнее состояние перед перезапуском), в которое перейдет CPU после запуска.



5.5.3 Трансферная карта

ВНИМАНИЕ
Защита карты памяти и слота от электростатического разряда Электростатические разряды могут повредить карту памяти или предусмотренный для нее слот в CPU. Для безопасного обращения с картой памяти следует убедиться, что выполняется одно или оба описанных ниже условия: • Пользователь стоит на заземленной, токопроводящей поверхности. • При работе с картой памяти пользователь носит заземленный браслет. Карта памяти должна храниться в токопроводящем контейнере.

Создание трансферной карты

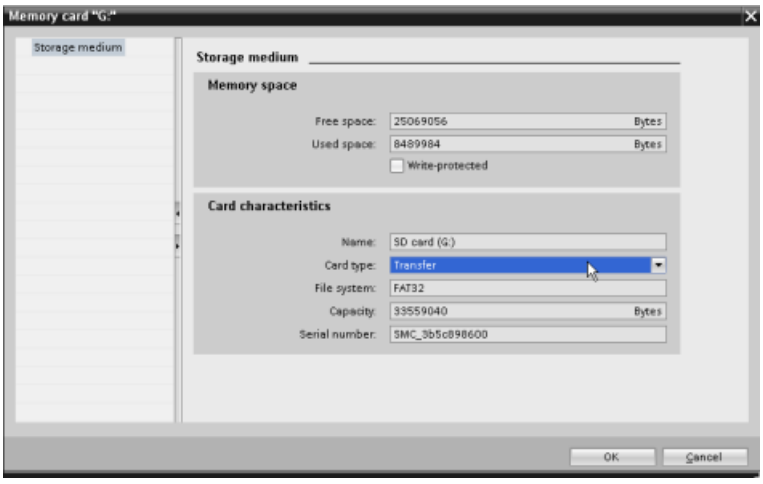
Следует помнить, что сначала необходимо настроить параметры запуска CPU (Страница 139), и лишь после этого копировать программу на трансферную карту. Для создания трансферной карты необходимо выполнить следующее:

1. Вставить пустую карту памяти SIMATIC без защиты от записи в подключенный к компьютеру картридер для SD карт. (Если карта защищена от записи, то следует сдвинуть переключатель блокировки в сторону от позиции "Lock".)
   Если повторно используется карта памяти SIMATIC, которая уже содержит программу пользователя, архивы данных, рецепты или обновление прошивки, то **следует** удалить эти файлы перед использованием карты. Использовать Windows Explorer, чтобы вывести на экран содержимое карты памяти. Удалить файл "S7_JOB.S7S", а также все имеющиеся папки (напр., "SIMATIC.S7S", "FWUPDATE.S7S", "DataLogs" и "Recipes").

ВНИМАНИЕ
НЕ УДАЛЯТЬ с карты памяти скрытые файл "__LOG__" и "crdinfo.bin". Файлы "__LOG__" и "crdinfo.bin" необходимы для работы карты памяти. При удалении этих файлов становится невозможным дальнейшее использование этой карты памяти с CPU.

2. Открыть в дереве проекта (проектном представлении) папку "Картридеры SIMATIC" и выбрать свой картридер.
3. Кликнуть правой кнопкой мыши по букве диска карты памяти и выбрать в контекстном меню опцию "Свойства". Откроется диалоговое окно "Карта памяти".

4. Выбрать в диалоговом окне "Карта памяти" в выпадающем списке "Тип карты" опцию "Трансфер".
   STEP 7 создает пустую трансферную карту. Если создается пустая трансферная карта, так как был утерян пароль для CPU (Страница 149), то извлечь трансферную карту из картридера.



5. Добавить программу, выбрав в дереве проекта CPU (напр., PLC_1 [CPU 1214C DC/DC/DC]) и перетащив его мышью на карту памяти. (В качестве альтернативы можно скопировать CPU и вставить на карту памяти.) Копирование CPU на карту памяти открывает диалоговое окно "Загрузить предварительный просмотр".
6. В диалоговом окне "Загрузить предварительный просмотр" нажать кнопку "Загрузить", чтобы скопировать CPU на карту памяти.
7. После того, как в диалоговом окне появится сообщение, что CPU (программа) загружены без ошибок, нажать кнопку "Завершить".

Использование трансферной карты

⚠ ПРЕДУПРЕЖДЕНИЕ

Перед вставкой карты памяти следует убедиться, что CPU не управляет процессом.

Вставка карты памяти переводит CPU в рабочее состояние STOP, что может сказаться на выполнении онлайн-процесса или работе машины. Непредсказуемый ход процесса или работы машины может привести к тяжким телесным повреждениям и/или материальному ущербу.

Перед вставкой трансферной карты следует убедиться, что CPU находится в режиме STOP, а процесс - в безопасном состоянии.

Примечание

Не вставлять трансферную карту с программой V3.0 в варианты CPU более высоких версий.

Трансферные карты с версии 3.0 несовместимы с S7-1200 CPU более высоких версий. Вставка карты памяти, которая содержит программу V3.0, вызывает ошибку CPU.

Если была вставлена трансферная карта с неправильной версией программы, то следует извлечь карту, перевести CPU из STOP в RUN, выполнить восстановление исходного состояния памяти (MRES) или выключить и снова включить CPU. После того, как CPU будет выведен из состояния ошибки, можно загрузить правильную программу CPU.

Для передачи программы в CPU выполнить следующие действия:

1. Вставить трансферную карту в CPU (Страница 137). Если CPU находится в режиме RUN, то выполняется его переход в состояние STOP. Индикатор обслуживания (MAINT) мигает, показывая, что карта памяти должна быть проанализирована. Имеющаяся программа по прежнему остается в CPU.
2. Выключить и снова включить CPU, чтобы проанализировать карту памяти. Альтернативным методом для перезапуска CPU является переход из STOP в RUN или восстановление исходного состояния памяти (MRES) в STEP 7.
3. После перезапуска и CPU анализирует карту памяти и копирует программу во внутреннюю загружаемую память CPU.
   Светодиод RUN/STOP поочередно мигает зеленым и желтым цветом, указывая на процесс копирования программы. Переход индикатора RUN/STOP к свечению желтым цветом, а светодиода MAINT - к миганию желтым цветом, означает завершение процесса копирования. Теперь можно извлечь карту памяти.
4. Перезапустить CPU (выключив и включив или альтернативными методами), чтобы проанализировать новую программу, которая была передана во внутреннюю загружаемую память.

После CPU переходит в состояние для запуска (RUN или STOP), который был сконфигурирован для проекта.

Примечание

Следует удалить трансферную карту до перевода CPU в режим RUN.

5.5.4 Программная карта

ВНИМАНИЕ
Электростатические разряды могут повредить карту памяти или предусмотренный для нее слот в CPU. При работе с картой памяти следует стоять на электропроводящей, заземленной поверхности и/или носить заземленный браслет. Карта памяти должна храниться в токопроводящем контейнере.



Убедиться, что карта памяти не защищена от записи. Для этого сдвинуть переключатель блокировки в сторону от позиции "Lock".

Перед копированием элементов программы на программную карту следует удалить с карты все ранее сохраненные на ней программы.

Создание программной карты

Карта памяти, если она используется в качестве программной карты, является внешней загружаемой памятью CPU. После извлечения программной карты внутренняя загружаемая память CPU будет пустой.

Примечание

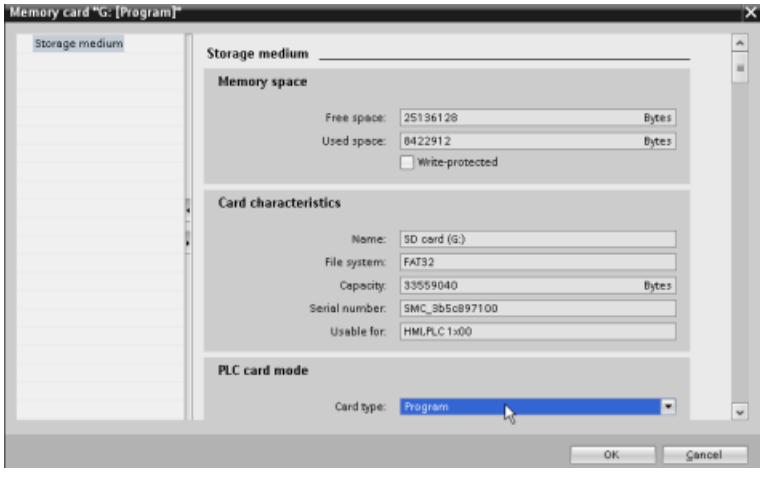
Если вставить пустую карту памяти в CPU и выполнить ее анализ, выключив и включив CPU, выполнив переход из STOP в RUN или восстановление исходного состояния памяти (MRES), то программа и принудительно задаваемые значения из внутренней загружаемой памяти CPU скопируются на карту памяти. (Карта памяти теперь является программной картой.) По окончании копирования программа во внутренней загрузочной памяти CPU стирается. После CPU переходит в сконфигурированное рабочее состояние для запуска (RUN или STOP).

Следует помнить, что сначала необходимо настроить параметры запуска CPU (Страница 139), и лишь после этого копировать проект на программную карту. Для создания программной карты необходимо выполнить следующее:

1. Вставить пустую карту памяти SIMATIC без защиты от записи в подключенный к компьютеру картридер для SD карт. (Если карта защищена от записи, то следует сдвинуть переключатель блокировки в сторону от позиции "Lock".)
   Если повторно используется карта памяти SIMATIC, которая уже содержит программу пользователя, архивы данных, рецепты или обновление прошивки, то **следует** удалить эти файлы перед использованием карты. С помощью Windows Explorer можно отобразить содержание карты памяти на экране и при необходимости удалить следующие файлы и папки:
   - S7_JOB.S7S
   - SIMATIC.S7S
   - FWUPDATE.S7S
   - DataLogs
   - Recipes

ВНИМАНИЕ
НЕ УДАЛЯТЬ с карты памяти скрытые файл "__LOG__" и "crdinfo.bin". Файлы "__LOG__" и "crdinfo.bin" необходимы для работы карты памяти. При удалении этих файлов становится невозможным дальнейшее использование этой карты памяти с CPU.

2. Открыть в дереве проекта (проектном представлении) папку "Картридеры/память USB" и выбрать свой картридер.
3. Кликнуть правой кнопкой мыши по букве диска карты памяти и выбрать в контекстном меню опцию "Свойства". Откроется диалоговое окно "Карта памяти".
4. Выбрать в диалоговом окне "Карта памяти" в контекстном меню опцию "Программа".



5. Добавить программу, выбрав в дереве проекта CPU (напр., PLC_1 [CPU 1214C DC/DC/DC]) и перетащив его мышью на карту памяти. (В качестве альтернативы можно скопировать CPU и вставить на карту памяти.) Копирование CPU на карту памяти открывает диалоговое окно "Загрузить предварительный просмотр".
6. В диалоговом окне "Загрузить предварительный просмотр" нажать кнопку "Загрузить", чтобы скопировать CPU на карту памяти.
7. После того, как в диалоговом окне появится сообщение, что CPU (программа) загружены без ошибок, нажать кнопку "Завершить".

Использование программной карты как внешней загружаемой памяти для CPU

⚠ ПРЕДУПРЕЖДЕНИЕ

Риски при вставке программной карты

Перед вставкой карты памяти следует убедиться, что CPU не управляет процессом.

Вставка карты памяти переводит CPU в рабочее состояние STOP, что может сказаться на выполнении онлайн-процесса или работе машины. Непредсказуемый ход процесса или работы машины может привести к тяжким телесным повреждениям и/или материальному ущербу.

Перед вставкой карты памяти следует убедиться, что CPU находится в режиме офлайн и в безопасном состоянии.

Для использования программной карты в CPU необходимо выполнить следующее:

1. Вставить программную карту в CPU. Если CPU находится в рабочем состоянии RUN, то он переходит в STOP. Индикатор обслуживания (MAINT) мигает, показывая, что карта памяти должна быть проанализирована.
2. Выключить и снова включить CPU, чтобы проанализировать карту памяти. Альтернативным методом для перезапуска CPU является переход из STOP в RUN или восстановление исходного состояния памяти (MRES) в STEP 7.
3. После перезапуска CPU и анализа программной карты CPU очищает внутреннюю загружаемую память CPU.

После CPU переходит в состояние для запуска (RUN или STOP), который был сконфигурирован для CPU.

Программная карта должна оставаться вставленной в CPU. Извлечение программной карты означает, что у CPU больше не будет программы во внутренней загружаемой памяти.

⚠ ПРЕДУПРЕЖДЕНИЕ

Риски при извлечении программной карты

При извлечении программной карты CPU теряет свою внешнюю загружаемую память и генерирует ошибку. После CPU переходит в рабочее состояние STOP и мигает индикатор ошибки.

Устройства управления могут выходить из строя в небезопасных рабочих состояниях, вызывая тем самым непредсказуемое поведение управляемого оборудования. Это приводит к непредсказуемому поведению системы автоматизации и может вызвать тяжкие телесные повреждения и/или материальный ущерб.

Можно извлечь программную карту, только если есть понимание того, что при этом из CPU удаляется программа.

Срок службы карты памяти SIMATIC

Срок службы карты памяти SIMATIC зависит от следующих факторов:

- Количество процессов стирания и записи на блок памяти
- Количество записанных байтов
- Внешние влияния, напр., температура окружающей среды

Примечание

Влияние процессов записи и стирания на срок службы карты памяти SIMATIC

Процессы записи и стирания, в первую очередь циклические, сокращают срок службы карты памяти SIMATIC.

Циклическое выполнение следующих инструкций сокращает срок службы карты памяти, в зависимости от числа процессов записи и данных:

- CREATE_DB (с ATTRIB "Создать DB в загружаемой памяти")
- DataLogWrite
- RecipeExport
- RecipeImport (если целевой DB находится в загружаемой памяти)
- WRIT_DBL
- SET_TIMEZONE

Наряду с циклическими процессами записи и стирания, запись и стирание очень больших объемов данных также негативно сказывается на сроке службы карты памяти SIMATIC.

5.5.5 Обновление прошивки

Карта памяти SIMATIC позволяет выполнить обновление для прошивки.

ВНИМАНИЕ
Защита карты памяти и слота от электростатического разряда Электростатические разряды могут повредить карту памяти или предусмотренный для нее слот в CPU. При работе с картой памяти следует стоять на электропроводящей, заземленной поверхности и/или носить заземленный браслет. Карта памяти должна храниться в токопроводящем контейнере.

Можно использовать карту памяти SIMATIC для загрузки обновлений прошивки с сайта поддержки пользователей Siemens Industry (http://support.industry.siemens.com). Перейти на этом сайте к "Загрузкам". Отсюда следует перейти к определенному типу модуля, который необходимо обновить.

В качестве альтернативы можно вызвать Веб-сайт S7-1200 для обновлений (https://support.industry.siemens.com/cs/ww/en/ps/13683/dl) напрямую.

Примечание

Нельзя обновить S7-1200 CPU V3.0 или более старой версии до прошивки S7-1200 V4.0 или V4.1.

Обновление прошивки может быть выполнено и одним из следующих способов:

- С помощью Интерактивных и диагностических инструментов STEP 7 (Страница 1394)
- Через стандартную веб-страницу "Информация о модулях" веб-сервера (Страница 1046)
- С помощью SIMATIC Automation Tool (https://support.industry.siemens.com/cs/ww/en/view/98161300)

ВНИМАНИЕ
Нельзя использовать стандартную утилиту Windows или любую другую утилиту для форматирования карты памяти Если карта памяти Siemens будет переформатирована, используя утилиту Microsoft Windows, то карта памяти станет непригодной для использования в CPU S7-1200.

Чтобы загрузить обновление прошивки на карту памяти, необходимо выполнить следующее:

1. Вставить пустую карту памяти SIMATIC без защиты от записи в подключенный к компьютеру картридер для SD карт. (Если карта защищена от записи, то сдвинуть переключатель блокировки в сторону от позиции "Lock".)
   Можно повторно использовать карту памяти SIMATIC, уже содержащую программу пользователя или другое обновление прошивки, но необходимо удалить некоторые файлы на карте памяти.
   Для повторного использования карты памяти перед загрузкой обновления прошивки **следует** удалить файл "S7_JOB.S7S", а также все имеющиеся папки с архивами данных и все каталоги (напр., "SIMATIC.S7S" или "FWUPDATE.S7S"). С помощью Windows Explorer можно отобразить содержание карты памяти на экране и удалить файлы и папки.

   ВНИМАНИЕ

   НЕ УДАЛЯТЬ с карты памяти скрытые файл "__LOG__" и "crdinfo.bin".

   Файлы "__LOG__" и "crdinfo.bin" необходимы для работы карты памяти. При удалении этих файлов становится невозможным дальнейшее использование этой карты памяти с CPU.

2. Выбрать Zip файл соответствующего модулю обновления прошивки и загрузить этот Zip файл на компьютер. Дважды кликнуть по файлу обновления, установить в качестве цели корневой каталог карты памяти SIMATIC и запустить процесс распаковки. После того, как распаковка завершена, корневой каталог (папка) карты памяти будет содержать каталог "FWUPDATE.S7S" и файл "S7_JOB.S7S".
3. Осторожно извлечь карту из картридера.

Для установки обновления прошивки выполнить следующие действия:

⚠ ПРЕДУПРЕЖДЕНИЕ

Перед установкой обновления прошивки необходимо убедиться, что CPU не управляет активным процессом.

Установка обновления прошивки переводит CPU в рабочее состояние STOP, что может сказаться на выполнении онлайн-процесса или работе машины. Непредсказуемый ход процесса или работы машины может привести к тяжким телесным повреждениям и/или материальному ущербу.

Перед вставкой карты памяти следует убедиться, что CPU находится в режиме офлайн и в безопасном состоянии.

1. Вставить карту памяти в CPU. Если CPU находится в рабочем состоянии RUN, то он переходит в STOP. Индикатор обслуживания (MAINT) мигает, показывая, что карта памяти должна быть проанализирована.
2. Перезагрузить CPU, чтобы запустить обновление прошивки. Альтернативным методом для перезапуска CPU является переход из STOP в RUN или восстановление исходного состояния памяти (MRES) в STEP 7.

   Примечание

   Чтобы завершить обновление прошивки модуля, следует обеспечить наличие бесперебойного внешнего питания 24 В DC для модуля.

   После перезапуска CPU начинается обновление прошивки. Светодиод RUN/STOP поочередно мигает зеленым и желтым цветом, указывая на процесс копирования обновления. Переход индикатора RUN/STOP к свечению желтым цветом, а светодиода MAINT - к миганию, означает завершение процесса копирования. Теперь необходимо извлечь карту памяти.
3. После извлечения карты памяти следует перезагрузить CPU (путем выключения/включения или альтернативными методами перезагрузки), чтобы загрузить новую прошивку.

Обновление прошивки не затрагивает программу пользователя и конфигурацию оборудования. После включения CPU переходит в сконфигурированный пусковой режим. (Если в качестве пускового режима для CPU был выбран "Горячий пуск - режим работы до СЕТЬ ВЫКЛ", то CPU находится в рабочем состоянии STOP, так STOP был последним рабочим режимом CPU.)

Примечание

Обновление нескольких подключенных к одному CPU модулей

Если в конфигурации оборудования есть несколько модулей, поддерживающих обновление прошивки с карты памяти, то CPU обновляет все затронутые модули (CM, SM и SB) в порядке конфигурирования, т.е. в порядке увеличения позиции модуля в конфигурации устройства в STEP 7.

Если загрузить несколько обновлений прошивки для нескольких модулей на карту памяти, то CPU выполнит обновления в порядке, в котором они были загружены на карту памяти.

5.6 Порядок действий при утерянном пароле

Если пароль для защищенного паролем CPU был утерян, то следует использовать пустую трансферную карту для удаления защищенной паролем программы. Пустая трансферная карта стирает внутреннюю загружаемую память CPU. После новая программа пользователя может быть загружена из STEP 7 в CPU.

Информацию о создании и использовании пустой трансферной карты можно найти в разделе Трансферные карты (Страница 140).

⚠ ПРЕДУПРЕЖДЕНИЕ

Перед вставкой карты памяти следует убедиться, что CPU не управляет процессом

При вставке трансферной карты в работающий CPU, он переходит в состояние STOP. Устройства управления могут выходить из строя в небезопасных рабочих состояниях, вызывая тем самым непредсказуемое поведение управляемого оборудования. Это приводит к непредсказуемому поведению системы автоматизации и может вызвать тяжкие телесные повреждения и/или материальный ущерб.

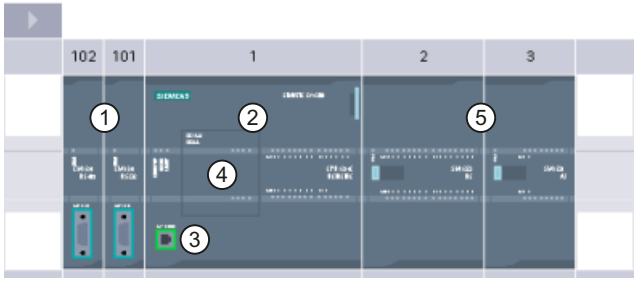
Перед вставкой трансферной карты следует убедиться, что CPU находится в режиме STOP, а процесс - в безопасном состоянии.

Следует удалить трансферную карту до перевода CPU в режим RUN.

Конфигурация устройства

6

Конфигурация устройства PLC создается путем добавления модуля CPU и других модулей к проекту.



① Коммуникационный модуль (CM) или коммуникационный процессор (CP): до 3, в слотах 101, 102 и 103

② CPU: слот 1

③ Порт PROFINET на CPU

④ Сигнальная плата (SB), коммуникационная плата (CB) или плата буферной батареи (BB): макс. 1, вставляется в CPU

⑤ Сигнальный модуль (SM) для цифровых или аналоговых входов/выходов: до 8, в слотах от 2 до 9

(8 у CPU 1214C, CPU 1215C и CPU 1217C, 2 у CPU 1212C, нет у CPU 1211C)

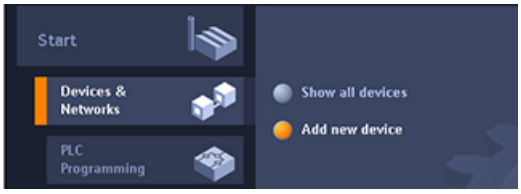
Управление конфигурацией

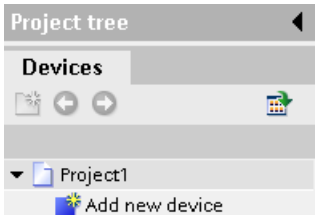
Конфигурация устройства для S7-1200 также поддерживает "Управление конфигурацией (Страница 156)", где можно настроить максимальную конфигурацию для проекта, включая модули, которые, возможно, не будут использоваться. Эта функция, иногда также называемая "управление опциями", позволяет настроить максимальную конфигурацию, которая позволяет использовать различные модули во множестве приложений.

6.1 Вставка CPU

Модуль CPU может быть вставлен в проект в портальном или проектном представлении STEP 7:

- Выбрать в портальном представлении портал "Устройства и сети" и кликнуть по "Добавить новое устройство".
- Двойной клик в проектном представлении под именем проекта по "Добавить новое устройство".





Убедиться, что выбрана правильная модель и версию прошивки из списка. При выборе модуля CPU в диалоговом окне "Добавление нового устройства" создается стойка и CPU.

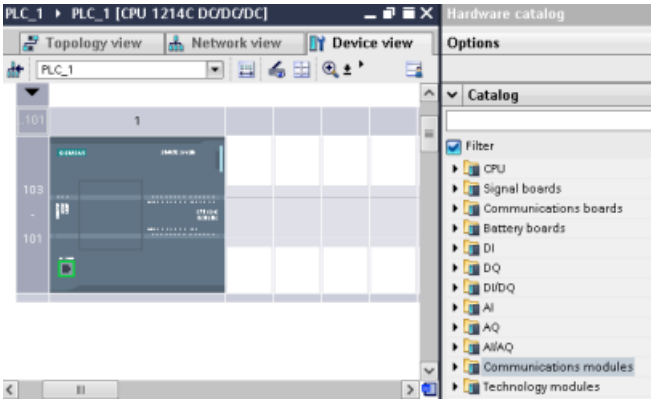
Примечание

В STEP 7 V14 и выше нельзя добавить в проект S7-1200 CPU версии 1.0.

Диалоговое окно "Добавление нового устройства"

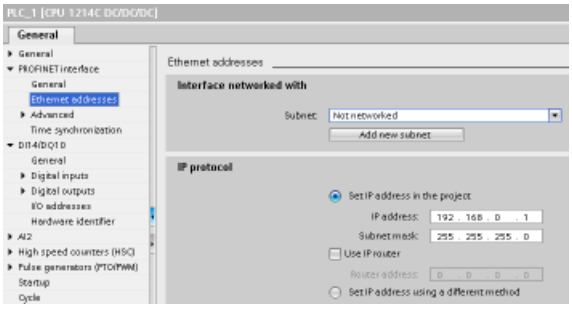


Просмотр устройства в конфигурации оборудования



После выбора CPU на вкладке оборудования его свойства отображаются в окне инспектора.

У CPU нет заранее сконфигурированного IP-адреса. Необходимо вручную назначить IP-адрес для CPU при создании конфигурации устройства. Если CPU подключен к маршрутизатору в сети, то следует ввести и IP-адрес для маршрутизатора.



6.2 Выгрузка конфигурации из подключенного CPU

STEP 7 предлагает два метода для выгрузки конфигурации оборудования из подключенного CPU:

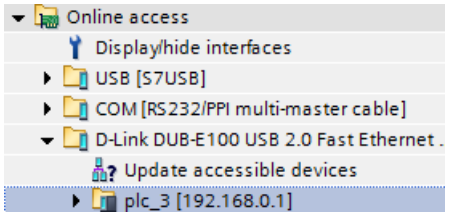
- Выгрузка подключенного устройства как новой станции
- Конфигурирование не специфицированного CPU посредством идентификации конфигурации оборудования подключенного CPU

Но следует помнить, что в первом случае выгружается не только конфигурация оборудования, но и программное обеспечение подключенного CPU.

Выгрузка устройства как новой станции

Для выгрузки подключенного устройства как новой станции, выполнить следующие шаги:

1. Раскрыть коммуникационный интерфейс в дереве проекта через узел "Интерактивный доступ"
2. Двойной клик по "Обновить доступные устройства".
3. Выбрать PLC из идентифицированных устройств.



4. Выбрать в интерактивном меню STEP 7 команду "Выгрузить устройство как новую станцию (аппаратная и программная часть)".

STEP 7 выгружает конфигурацию оборудования и программные блоки.

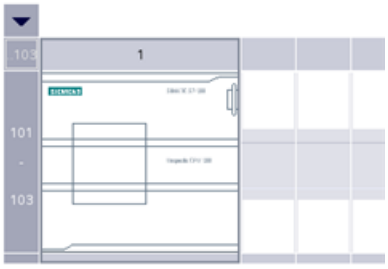
Идентификация конфигурации оборудования не специфицированного CPU

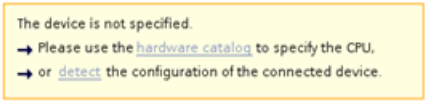


Если есть соединение с CPU, то можно выгрузить конфигурацию этого CPU, включая любые модули, из устройства в проект. Просто создать новый проект и выбрать "не специфицированный CPU" вместо определенного CPU. (Также можно полностью пропустить конфигурацию устройства, выбрав "Создать программу PLC" в процедуре "Первые шаги". STEP 7 в этом случае автоматически создает не специфицированный CPU.)

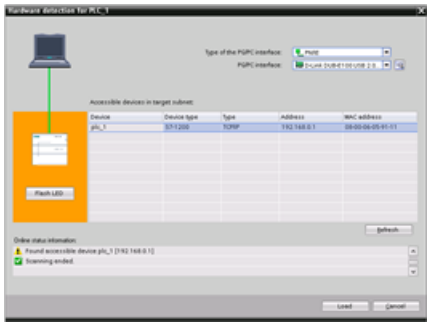
Выбрать в редакторе текстов программ в меню "Онлайн" команду "Идентификация оборудования".

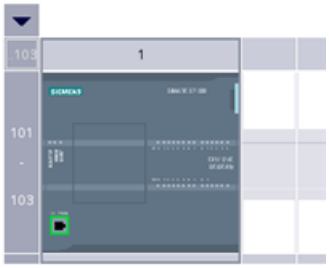
Выбрать в редакторе конфигурации устройства опцию для идентификации конфигурации подключенного устройства.





После выбора CPU в интерактивном диалоговом окне и нажатия кнопки загрузки, STEP 7 выгружает конфигурацию оборудования, включая возможные модули (SM, SB или CM), из CPU. После можно сконфигурировать параметры для CPU и модулей (Страница 167).





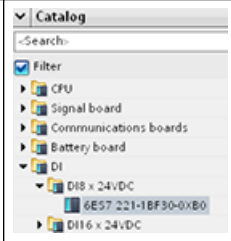
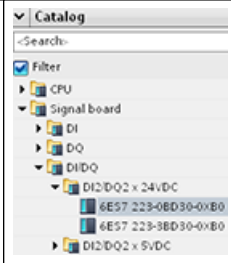
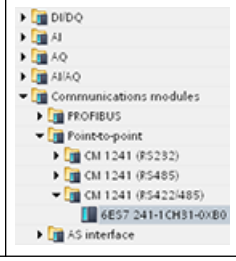
6.3 Добавление модулей в конфигурацию

Можно использовать каталог оборудования для добавления модулей к CPU:

- Сигнальные модули (SM) для добавления цифровых или аналоговых входов и выходов. Эти модули подключаются с правой стороны от модуля CPU.
- Сигнальные платы (SB) предлагают лишь ограниченное количество дополнительных входов/выход для CPU. SB вставляется с лицевой стороны модуля CPU.
- Плата буферной батареи 1297 (BB) обеспечивает долговременную буферизацию часов реального времени. BB вставляется с лицевой стороны модуля CPU.
- Коммуникационные платы (CB) предлагают дополнительный порт передачи данных (напр., RS485). CB вставляется с лицевой стороны модуля CPU.
- Коммуникационные модули (CM) и коммуникационные процессоры (CP) предлагают дополнительный порт передачи данных, напр., для PROFIBUS или GPRS. Эти модули подключаются с левой стороны от модуля CPU.

Для вставки модуля в конфигурацию устройства, выбрать его в каталоге оборудования и дважды кликнуть по нему или перетащить модуль в отмеченный слот. Необходимо включить модули в конфигурацию устройства и загрузить конфигурацию оборудования в CPU, чтобы сделать модули готовыми к работе.

Таблица 6-1 Добавление модуля в конфигурацию устройства

Модуль	Выбор модуля	Вставка модуля	Результат
SM			
SB, BB или CB			
CM или CP			

С помощью функции "Управления конфигурацией" (Страница 156) можно добавлять сигнальные модули и сигнальные платы в конфигурацию устройства, которая может не соответствовать фактическому оборудованию для определенного приложения, но она будет работать в других приложениях, которые используют ту же программу пользователя, модель CPU и, возможно, некоторые из сконфигурированных модулей.

6.4 Управление конфигурацией

6.4.1 Преимущества и использование управления конфигурацией

Управление конфигурацией может быть полезным инструментом при создании решения автоматизации, которое будет использоваться в различных вариантах на нескольких установках.

Конфигурация устройства STEP 7 и программа пользователя могут быть загружены в различные установленные конфигурации PLC. Потребуются лишь незначительные изменения проекта STEP 7 для его адаптации к новой конфигурации.

6.4.2 Конфигурирование центральной системы и дополнительных модулей

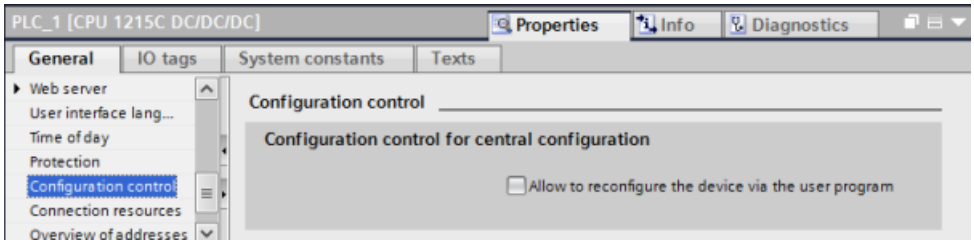
Управление конфигурацией в STEP 7 и S7-1200 позволяет сконфигурировать максимальную конфигурацию для стандартной системы с различными версиями (опциями), которые используют разновидности этой конфигурации. В справочнике PROFINET с STEP 7 (http://support.automation.siemens.com/WW/view/en/49948856) проекты такого типа называются "проектами для серийного оборудования".

Набор управляющих данных, программируемый в пусковом OB, уведомляет CPU относительно тех модулей, которые отсутствуют в реальной системе по сравнению с конфигурацией, или какие модули расположены в других слотах по сравнению с конфигурацией. Управление конфигурацией не оказывает влияние на назначение параметров модулей.

Управление конфигурацией обеспечивает гибкость при изменении компоновки системы, пока реальная конфигурация является частью максимальной конфигурации устройства в STEP 7.

Чтобы активировать управление конфигурацией и структурировать требуемый набор управляющих данных, выполнить следующие шаги:

1. Как опция можно сбросить CPU на заводские установки, чтобы исключить наличие несовместимого набора управляющих данных в CPU.
2. Выбрать CPU в конфигурации устройства в STEP 7.
3. Активировать опцию "Переконфигурирование устройства через программу пользователя" узла "Управление конфигурацией" в свойствах CPU.

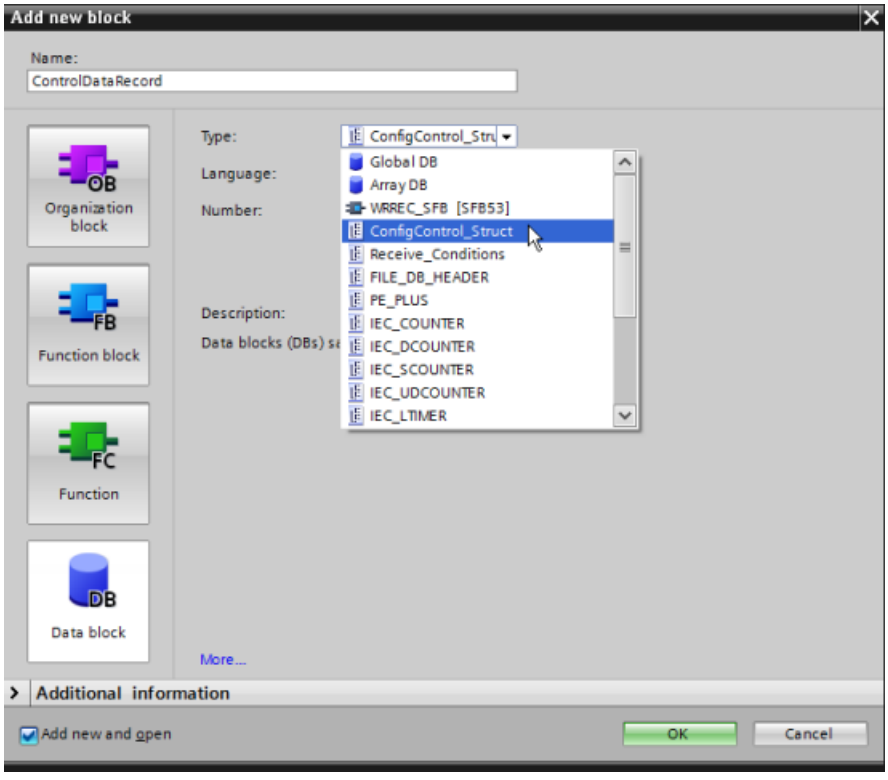


4. Создать тип данных PLC для набора управляющих данных. Сконфигурировать его как структуру (Struct) с четырьмя USint для данных управления конфигурацией и дополнительными USint согласно числу слотов для максимальной конфигурации S7-1200 следующим образом:

ConfigControl_Struct

	Name	Data type	Default value	Comment
1	ConfigControl	Struct		
2	Block_length	USInt	16	Length of control data record, including header
3	Block_ID	USInt	196	Data record number
4	Version	USInt	5	
5	Subversion	USInt	0	
6	Slot_1	USInt	255	Assignment for CPU annex card/Actual annex card
7	Slot_2	USInt	255	Configured slot 2/ Assigned "real" slot
8	Slot_3	USInt	255	Configured slot 3 / Assigned "real" slot
9	Slot_4	USInt	255	Configured slot 4 / Assigned "real" slot
10	Slot_5	USInt	255	Configured slot 5 / Assigned "real" slot
11	Slot_6	USInt	255	Configured slot 6 / Assigned "real" slot
12	Slot_7	USInt	255	Configured slot 7 / Assigned "real" slot
13	Slot_8	USInt	255	Configured slot 8 / Assigned "real" slot
14	Slot_9	USInt	255	Configured slot 9 / Assigned "real" slot
15	Slot_101	USInt	255	Configured slot 101 / Assigned "real" slot
16	Slot_102	USInt	255	Configured slot 102 / Assigned "real" slot
17	Slot_103	USInt	255	Configured slot 103 / Assigned "real" slot

5. Создать блок данных с выбранным типом данных PLC.



6. Сконфигурировать в этом блоке данных параметры Block_length, Block_ID, Version и Subversion как показано ниже. Сконфигурировать значения для слотов на основании их наличия или отсутствия и позиции в фактической компоновке:
   - 0: Сконфигурированный модуль отсутствует в фактической конфигурации. (Слот пустой.)
   - От 1 до 9, от 101 до 103: Фактическая позиция сконфигурированного слота
   - 255: Конфигурация устройства STEP 7 не содержит модуля в этом слоте.

Примечание

Управление конфигурацией недоступно для HSC и PTO на сигнальной плате

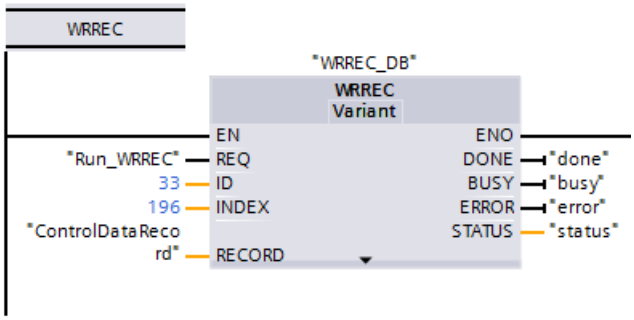
Если в CPU есть сигнальная плата, сконфигурированная для HSC или PTO, то нельзя отключить эту сигнальную плату через "0" в Слоте_1 набора управляющих данных для конфигурации. Сконфигурированные устройства HSC и PTO являются обязательными компонентами CPU для управления конфигурацией.

ControlDataRecord

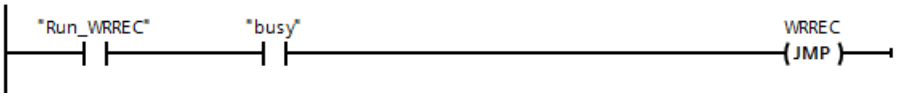
	Name	Data type	Start value	Comment
1	Static			
2	ConfigControl	Struct		
3	Block_length	USInt	16	Length of control data record, including header
4	Block_ID	USInt	196	Data record number
5	Version	USInt	5	
6	Subversion	USInt	0	
7	Slot_1	USInt	255	Assignment for CPU annex card/Actual annex c...
8	Slot_2	USInt	255	Configured slot 2/ Assigned "real" slot
9	Slot_3	USInt	255	Configured slot 3 / Assigned "real" slot
10	Slot_4	USInt	255	Configured slot 4 / Assigned "real" slot
11	Slot_5	USInt	255	Configured slot 5 / Assigned "real" slot
12	Slot_6	USInt	255	Configured slot 6 / Assigned "real" slot
13	Slot_7	USInt	255	Configured slot 7 / Assigned "real" slot
14	Slot_8	USInt	255	Configured slot 8 / Assigned "real" slot
15	Slot_9	USInt	255	Configured slot 9 / Assigned "real" slot
16	Slot_101	USInt	255	Configured slot 101 / Assigned "real" slot
17	Slot_102	USInt	255	Configured slot 102 / Assigned "real" slot
18	Slot_103	USInt	255	Configured slot 103 / Assigned "real" slot

Назначение значений для слотов см. Пример для управления конфигурацией (Страница 163).

7. Вызвать в пусковом OB расширенную инструкцию WRREC (запись набора данных), чтобы перенести созданный набор управляющих данных на индекс 196 идентификатора оборудования 33. Использовать метку перехода (JMP) для ожидания выполнения инструкции WRREC.
   Сегмент 1:



Сегмент 2:



Примечание

Управление конфигурацией активируется только после того, как инструкция WRREC перенесла в пусковой OB набор управляющих данных. Если управление конфигурацией активировано, а в CPU отсутствует набор управляющих данных, то CPU перейдет в режим STOP сразу после запуска. Поэтому важно, чтобы пусковой OB был запрограммирован для передачи набора управляющих данных.

Расположение модулей

В следующей таблице представлено назначение слотов:

Слот	Модули
1	Сигнальная плата или коммуникационная плата (дополнительная карта CPU)
От 2 до 9	Сигнальные модули
От 101 до 103	Коммуникационные модули

Набор управляющих данных

Набор управляющих данных 196 содержит назначения слотов и фактическую конфигурацию, представленную ниже:

Байт	Элемент	Значение	Пояснение
0	Длина блока	16	Заголовок
1	ID блока	196	
2	Версия	5	
3	Подверсия	0	

Байт	Элемент	Значение	Пояснение
4	Назначение дополнительной карты CPU	Фактическая дополнительная карта, 0 или 255*	Элемент управления Описывает в каждом элементе, какой из реальных слотов устройства назначен сконфигурированному слоту.
5	Назначение сконфигурированного слота 2	Фактический слот, 0 или 255*	
...	...	...	
12	Назначение сконфигурированного слота 9	Фактический слот, 0 или 255*	
13	Назначение сконфигурированного слота 101	Фактический слот или 255*	В отличие от сигнальных модулей, фактический слот для физически существующих коммуникационных модулей должен совпадать со сконфигурированным слотом.
14	Назначение сконфигурированного слота 102	Фактический слот или 255*	
15	Назначение сконфигурированного слота 103	Фактический слот или 255*	

***Значения для слотов:**

0: Сконфигурированный модуль отсутствует в фактической конфигурации. (Слот пустой.)

От 1 до 9, от 101 до 103: Фактическая позиция сконфигурированного слота

255: Конфигурация устройства STEP 7 не содержит модуля в этом слоте.

Примечание

Альтернатива созданию типа переменных (тегов) PLC

В качестве альтернативы созданию пользовательского типа переменных (тегов) PLC можно создать блок данных со всеми структурными элементами набора управляющих данных. Также можно сконфигурировать несколько структур в этом блоке данных в качестве различных конфигураций набора управляющих данных. Любой из вариантов является эффективным способом передачи набора управляющих данных в процессе запуска.

Правила

Придерживаться следующих правил:

- Управление конфигурацией не поддерживает изменения позиции для коммуникационных модулей. Также нельзя использовать управление конфигурацией для отключения коммуникационных модулей. Позиции слотов, указанные в наборе управляющих данных для слотов 101 - 103, должны соответствовать фактической конфигурации. Если для слота в конфигурации устройства не сконфигурирован модуль, то ввести для этой позиции значение 255 в наборе управляющих данных. Если для слота сконфигурирован модуль, то указать сконфигурированный слот как фактический слот для этой позиции.
- F-модули ввода-вывода не поддерживают управление конфигурацией. Позиции для F-модуля I/O в наборе управляющих данных должны соответствовать сконфигурированной позиции слота F-модуля I/O. При попытке перемещения или удаления сконфигурированного F-модуля I/O с набором управляющих данных, все фактически установленные F-модули I/O сигнализируют ошибку параметрирования и блокируют замену.

- Нельзя оставлять неиспользованные слоты между занятыми слотами. Если, к примеру, в фактической конфигурации есть модуль в слоте 4, то модули должны быть и в слотах 2 и 3. Если в фактической конфигурации есть коммуникационный модуль в слоте 102, то соответственно и в слоте 101 должен быть вставлен модуль.
- Если управление конфигурацией активировано, то CPU будет готов к работе после загрузки набора управляющих данных. Модуль CPU переходит из пуска в STOP, если пусковой OB не передает правильного набора управляющих данных. В этом случае CPU не инициализирует центральные I/O и причина перехода в STOP записывается в буфер диагностики.
- Модуль CPU помещает успешно переданный набор управляющих данных в сохраняющую память; это означает, что нет необходимости снова записывать набор управляющих данных 196 после перезапуска, если конфигурация не была изменена.
- Каждый реальный слот должен присутствовать только один раз в наборе управляющих данных.
- Только один реальный слот может быть назначен каждому сконфигурированному слоту.

Примечание

Изменение конфигурации

Запись набора управляющих данных с измененной конфигурацией вызывает следующую автоматическую реакцию CPU: восстановление исходного состояния памяти и перезапуск с измененной конфигурацией.

Модуль CPU удаляет исходный набор управляющих данных и помещает новый набор управляющих данных в сохраняющую память.

Поведение при работе

Для отображения онлайн и для отображения в буфере диагностики (модуль исправен или модуль неисправен), STEP 7 использует конфигурацию устройства, а не отличающуюся реальную конфигурацию.

Пример: Модуль выводит диагностические данные. Этот модуль сконфигурирован в слоте 4, но фактически вставлен в слот 3. Онлайн-представление показывает ошибку слота 4. В реальной конфигурации модуль в слоте 3 сигнализирует ошибку на светодиодном индикаторе.

Если в наборе управляющих данных модуль сконфигурирован как отсутствующий (0), то система автоматизации ведет себя следующим образом:

- Модули, определяемые как не существующие в наборе управляющих данных, не предоставляют диагностические данные, и их состояние всегда OK. Значение статуса OK.
- Прямой доступ по записи к выходам или доступ по записи к образу процесса выходов, которые отсутствуют, не имеет эффекта; CPU не сообщает о наличии ошибки доступа.
- Прямой доступ по записи к входам или доступ по записи к образу процесса входов, которые отсутствуют, вызывает значение "0" для каждого входа; CPU не сообщает о наличии ошибки доступа.

- Запись набора данных в модуль, который отсутствуют, не имеет эффекта; CPU не сообщает о наличии ошибки.
- При попытке чтения набора данных из отсутствующего модуля появляется сообщение об ошибке, так как CPU не может вернуть правильный набор данных.

Сообщения об ошибках

CPU возвращает следующие сообщения об ошибках, если ошибка возникает во время записи набора управляющих данных:

Код ошибки	Значение
16#80B1	Недопустимая длина; информация о длине в наборе управляющих данных неправильная.
16#80B5	Параметры управления конфигурацией не назначены
16#80E2	Набор данных был передан в неправильном контексте OB. Набор данных должен передаваться в пусковом OB.
16#80B0	Тип блока (байт 2) в наборе управляющих данных не 196
16#80B8	Ошибка параметра; модуль сигнализирует о недопустимых параметрах, например: • Набор управляющих данных пытается изменить конфигурацию коммуникационного модуля или дополнительной коммуникационной карты. Фактическая конфигурация для коммуникационных модулей и дополнительной коммуникационной карты должна совпадать с конфигурацией STEP 7. • Назначенное значение для несконфигурированного слота в проекте STEP 7 не равно 255. • Назначенное значение для сконфигурированного слота вне допустимого диапазона. • В назначенной конфигурации есть "внутренний" пустой слот, например, слот n назначен, а слот n-1 не назначен.

6.4.3 Пример для управления конфигурацией

Настоящий пример описывает конфигурацию, состоящую из модуля CPU и трех модулей ввода-вывода. Модуль в слоте 3 отсутствует в первой фактической конфигурации и будет "скрыт" с помощью управления конфигурацией.

Во второй конфигурации изначально скрытый модуль хотя и используется, но теперь сконфигурирован в последнем слоте. Измененный набор управляющих данных передает информацию о назначений модулей для слотов.

Пример: Фактическая компоновка со сконфигурированным, но не используемым модулем

Конфигурация устройства содержит все модули, которые могут присутствовать в реальной компоновке (максимальная конфигурация). В данном случае модуль, назначенный слоту 3 в конфигурации устройства, отсутствует в реальной компоновке.

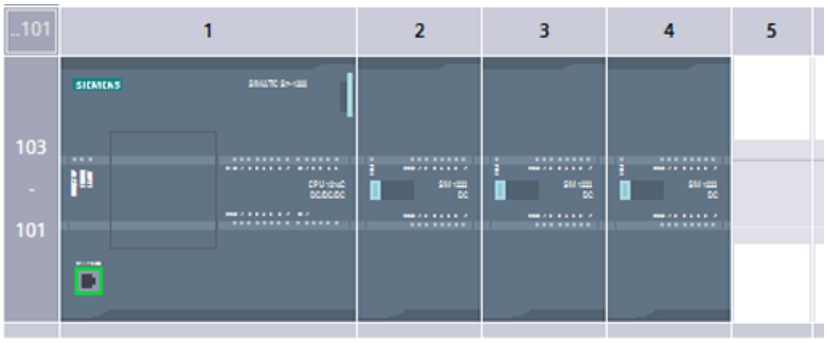


Рисунок 6-1 Конфигурация устройства для максимальной комплектации с тремя сигнальными модулями



Рисунок 6-2 Фактическая конфигурация с отсутствующим модулем для слота 3 и сконфигурированным для слота 4 модулем в слоте 3

Для указания на отсутствие модуля, необходимо задать 0 для слота 3 в наборе управляющих данных.

ControlDataRecord

	Name	Data type	Start value	Comment
1	Static			
2	ConfigControl	Struct		
3	Block_length	USInt	16	Length of control data record, including header
4	Block_ID	USInt	196	Data record number
5	Version	USInt	5	
6	Subversion	USInt	0	
7	Slot_1	USInt	255	Assignment for CPU annex card/Actual annex ca..
8	Slot_2	USInt	2	Configured slot 2 / Assigned "real" slot
9	Slot_3	USInt	0	Configured slot 3 / Assigned "real" slot
10	Slot_4	USInt	3	Configured slot 4 / Assigned "real" slot
11	Slot_5	USInt	255	Configured slot 5 / Assigned "real" slot
12	Slot_6	USInt	255	Configured slot 6 / Assigned "real" slot
13	Slot_7	USInt	255	Configured slot 7 / Assigned "real" slot
14	Slot_8	USInt	255	Configured slot 8 / Assigned "real" slot
15	Slot_9	USInt	255	Configured slot 9 / Assigned "real" slot
16	Slot_101	USInt	255	Configured slot 101 / Assigned "real" slot
17	Slot_102	USInt	255	Configured slot 102 / Assigned "real" slot
18	Slot_103	USInt	255	Configured slot 103 / Assigned "real" slot

Пример: Фактическая конфигурация с модулем, впоследствии добавляемым в другой слот

Во втором примере модуль в слоте 3 конфигурации устройства присутствует в реальной конфигурации, но находится в слоте 4.

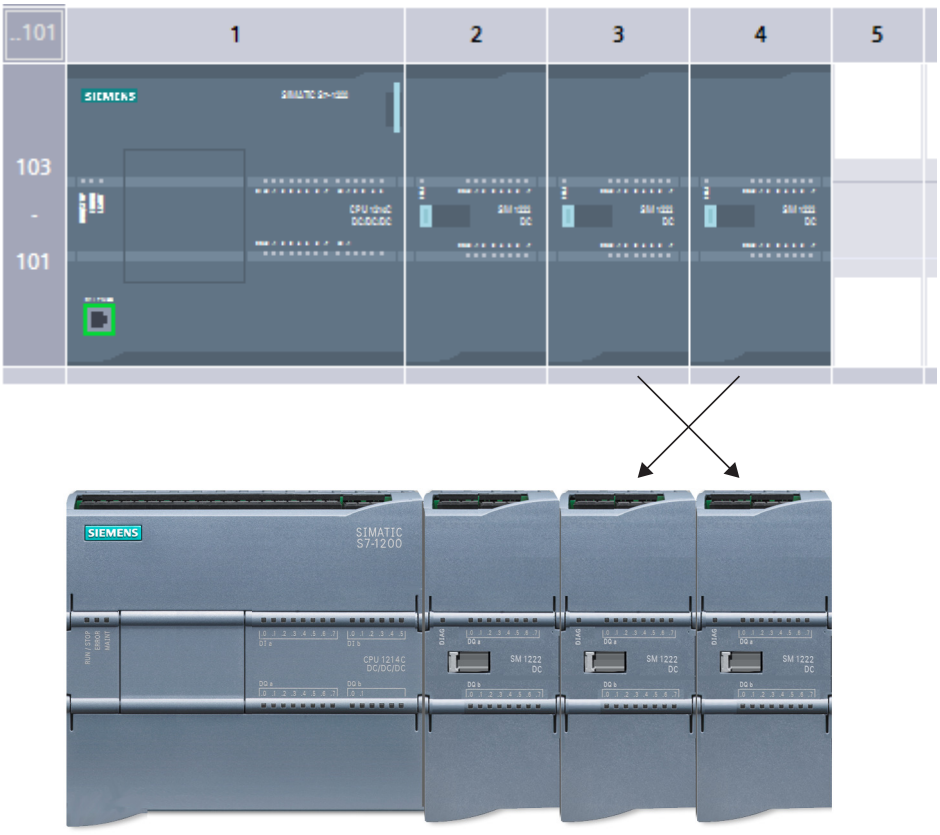


Рисунок 6-3 Модули в слотах 3 и 4 из конфигурации устройства поменяны местами в фактической конфигурации

Для того, что конфигурация устройства соответствовала фактической конфигурации, следует назначить модули на правильные слоты путем редактирования набора управляющих данных.

ControlDataRecord

	Name	Data type	Start value	Comment
1	Static			
2	ConfigControl	Struct		
3	Block_length	USInt	16	Length of control data record, including header
4	Block_ID	USInt	196	Data record number
5	Version	USInt	5	
6	Subversion	USInt	0	
7	Slot_1	USInt	255	Assignment for CPU annex card/Actual annex card
8	Slot_2	USInt	2	Configured slot 2/ Assigned "real" slot
9	Slot_3	USInt	4	Configured slot 3 / Assigned "real" slot
10	Slot_4	USInt	3	Configured slot 4 / Assigned "real" slot
11	Slot_5	USInt	255	Configured slot 5 / Assigned "real" slot
12	Slot_6	USInt	255	Configured slot 6 / Assigned "real" slot
13	Slot_7	USInt	255	Configured slot 7 / Assigned "real" slot
14	Slot_8	USInt	255	Configured slot 8 / Assigned "real" slot
15	Slot_9	USInt	255	Configured slot 9 / Assigned "real" slot
16	Slot_101	USInt	255	Configured slot 101 / Assigned "real" slot
17	Slot_102	USInt	255	Configured slot 102 / Assigned "real" slot
18	Slot_103	USInt	255	Configured slot 103 / Assigned "real" slot

6.5 Изменение устройства

Можно изменить тип устройства для сконфигурированного CPU или модуля. Кликнуть в конфигурации устройства правой кнопкой мыши по устройству и выбрать команду "Изменить устройство" из контекстного меню. В диалоговом окне выбрать CPU или модуль, который необходимо изменить. В диалоговом окне "Изменить устройство" отображается информация о совместимости для обоих затронутых устройств.

Дополнительную информацию об изменении версии CPU устройств можно найти здесь Замена CPU V3.0 на CPU V4.x (Страница 1655).

6.6 Конфигурирование работы CPU

6.6.1 Обзор

Для конфигурирования рабочих параметров CPU выбрать CPU в просмотре устройств (синяя рамка вокруг всего CPU) и после открыть вкладку "Свойства" в окне инспектора.

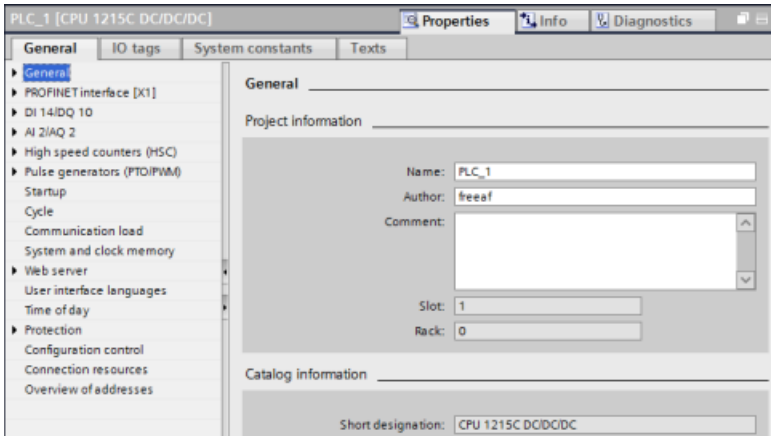


Таблица 6-2 Свойства CPU

Свойство	Описание
Интерфейс PROFINET	Установка IP-адреса для CPU и синхронизация времени
DI, DO и AI	Установка поведения локальных (встроенных) цифровых и аналоговых I/O (например, время фильтра для цифровых входов и реакция цифровых выходов на перевод CPU в рабочее состояние STOP).
Быстрые счетчики (Страница 557) и генераторов импульсов (Страница 492)	Активация и настройка высокоскоростных счётчиков (HSC) и генераторов импульсов для формирования последовательности импульсов (PTO) и широтно-импульсной модуляции (PWM) Если выходы CPU или сигнальной платы конфигурируются как генераторы импульсов (для PWM или инструкций управления перемещением), то соответствующие адреса выходов удаляются из памяти выходов и не могут использоваться для других целей в программе пользователя. Если программа пользователя записывает значение на выход, который используется как генератор импульсов, то CPU не записывает это значение на физический выход.

Свойство	Описание
Пуск (Страница 87)	**Пуск после СЕТЬ ВКЛ:** Установка поведения CPU после выключения и повторного включения, напр., для запуска в рабочем состоянии STOP или перехода в RUN после горячего пуска
	Совместимость оборудования: Конфигурация стратегии замены для всех системных компонентов (SM, SB, CM, CP и CPU): • Разрешает приемлемую замену • Разрешает любую замену (по умолчанию) У каждого модуля есть внутренние требования по совместимости при замене, учитывающие число I/O, электрическую совместимость и другие схожие точки сравнения. Пример: SM с 16 каналами может быть приемлемой заменой для SM с 8 каналами, но SM с 8 каналами не является приемлемой заменой для SM с 16 каналами. При выборе "Разрешает приемлемую замену", STEP 7 использует правила замены. В остальных случаях STEP 7 разрешает любую замену.
	Время назначения параметров для распределенных I/O: Определяет максимальный период времени (установка по умолчанию: 60000мс), для перехода распределенное периферии в онлайн-режим. (Во время запуска CPU на CM и CP подается напряжение и передаются параметры коммуникации. Это время назначения устанавливает период времени, за который должны перейти в режим онлайн модули I/O, подключенные к CM или CP). CPU переходит в состояние RUN, как только распределенная периферия перейдет в онлайн, независимо от времени назначения. Если за это время распределенная периферия не перешла в онлайн, то CPU все же переходит в режим RUN, но без распределенной периферии. **Примечание:** Если в конфигурации есть CM 1243-5 (PROFIBUS-Master), то нельзя задавать для этого параметра значения ниже 15 секунд (15000 мс), чтобы гарантировать возможность для перехода модуля в онлайн.
	OB должны поддерживать прерывание: Устанавливает, может ли обработка OB (всех OB) в CPU прерываться или нет (Страница 103).
Цикл (Страница 107)	Определение максимального времени цикла или фиксированного минимального времени цикла.
Коммуникационная нагрузка	Выделение процентной доли времени CPU для задач коммуникации
Системная и тактовая память (Страница 111)	Установка байта для функций системной памяти и установка байта для функций тактовой памяти (при этом каждый бит переключается с заданной частотой).
Веб-сервер (Страница 1023)	Активация и конфигурирование функции веб-сервера
Время суток	Выбор часового пояса и установка летнего/зимнего времени
Многоязыковая поддержка (Страница 172)	Назначение языка проекта для веб-сервера. Этот язык используется для отображения сообщений из буфера диагностики на интерфейсе пользователя веб-сервера.
Защита (Страница 203)	Установка защит от чтения/записи и паролей для доступа к CPU
Управление конфигурацией (Страница 156)	Обеспечивает создание базовой конфигурации, которая может изменяться в соответствии с конкретной фактической конфигурацией устройства.
Ресурсы подключения (Страница 807)	Обзор ресурсов коммуникационных соединений CPU и числа сконфигурированных ресурсов подключения.
Обзор адресов	Обзор адресов I/O, сконфигурированных для CPU.

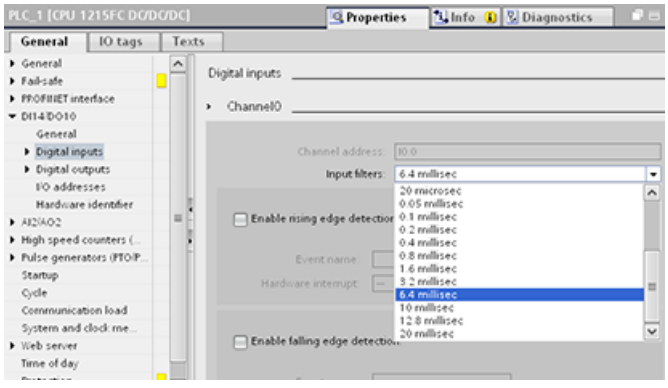
6.6.2 Установка времени фильтра для цифровых входов

Фильтры цифровых входов защищают программу от отклика на нежелательные быстрые изменения входных сигналов, вызванные, к примеру, дребезгом контактов или электрическими помехами. Время фильтра по умолчанию в 6,4 мс блокирует нежелательные действия схемы от типовых механических контактов. Различные функции в приложении могут потребовать более короткого времени фильтра для обнаружения и отклика на сигналы от быстрых датчиков или более длительного времени фильтра, чтобы блокировать медленный дребезг контакта или более длительный импульсный шум.

Время фильтра для входа в 6,4 мс означает, что одиночное изменение сигнала из ‘0’ в ‘1’ или из ‘1’ в ‘0’, должно продолжаться приблизительно 6,4 мс, чтобы быть обнаруженным, и одиночный импульс высокого или низкого уровня короче, чем приблизительно 6,4 мс, не обнаруживается. Если входной сигнал переключается между ‘0’ и ‘1’ быстрее, чем время фильтра, то значение входа может измениться в программе пользователя, если общая продолжительность импульсов с новыми значениями превысит время фильтра по старым значениям.

Фильтр для цифровых входов работает следующим образом:

- При вводе "1" начинается отсчет времени на увеличение; он останавливается при достижении времени фильтра. При достижении времени фильтра точка в образце процесса изменяется с "0" на "1".
- При вводе "0" начинается отсчет времени на уменьшение и останавливается при достижении "0". При достижении значения "0" точка в образце процесса изменяется с "1" на "0".
- При колебании значения на входе происходит частичный отсчет времени на увеличение и уменьшение. Образ процесс изменяется, когда сумма-нетто подсчитанных значений достигает либо времени фильтра, либо "0".
- Быстро изменяющийся сигнал с большим количеством "0", чем "1" возможно перейдет в "0", а если будет больше "1", чем "0", то регистр образа процесса возможно изменится в "1".



Каждый вход имеет одну конфигурацию фильтра, которая подходит для любых целей: технологические входы, прерывания, захват импульсов и входы высокоскоростных счетчиков. Для установки времени для входного фильтра выбрать "Цифровые входы".

Время фильтра по умолчанию для цифровых входов составляет 6,4 мс. Можно выбрать время фильтра из выпадающего списка "Фильтр для входа" Допустимый диапазон времен фильтра составляет от 0,1 мкс до 20,0 мс.

⚠ ПРЕДУПРЕЖДЕНИЕ

Риски, связанные с изменениями времени фильтра для каналов цифрового ввода

Если время фильтра для канала цифрового ввода изменено по сравнению с предыдущей настройкой, то для нового входного значения может потребоваться удержание уровня "0" в течение до 20,0 мс, чтобы фильтр мог полноценно реагировать на новые входы. В течение этого времени может случиться, что короткие с уровнем "0" продолжительностью менее 20,0 мс не будут быть обнаружены или подсчитаны.

Изменение времени фильтра может вызвать непредсказуемое поведение оборудования и, как следствие, привести к тяжким телесным повреждениям и/или материальному ущербу.

Чтобы гарантировать, что новое время фильтра сразу вступает в силу, необходимо выключить и снова включить CPU.

Конфигурация времени фильтра для используемых в качестве HSC цифровых входов

Для входов, используемых как высокоскоростные счётчики (HSC), необходимо установить время фильтра для входа на подходящее значение для исключения ошибок счета.

Siemens рекомендует использовать следующие установки:

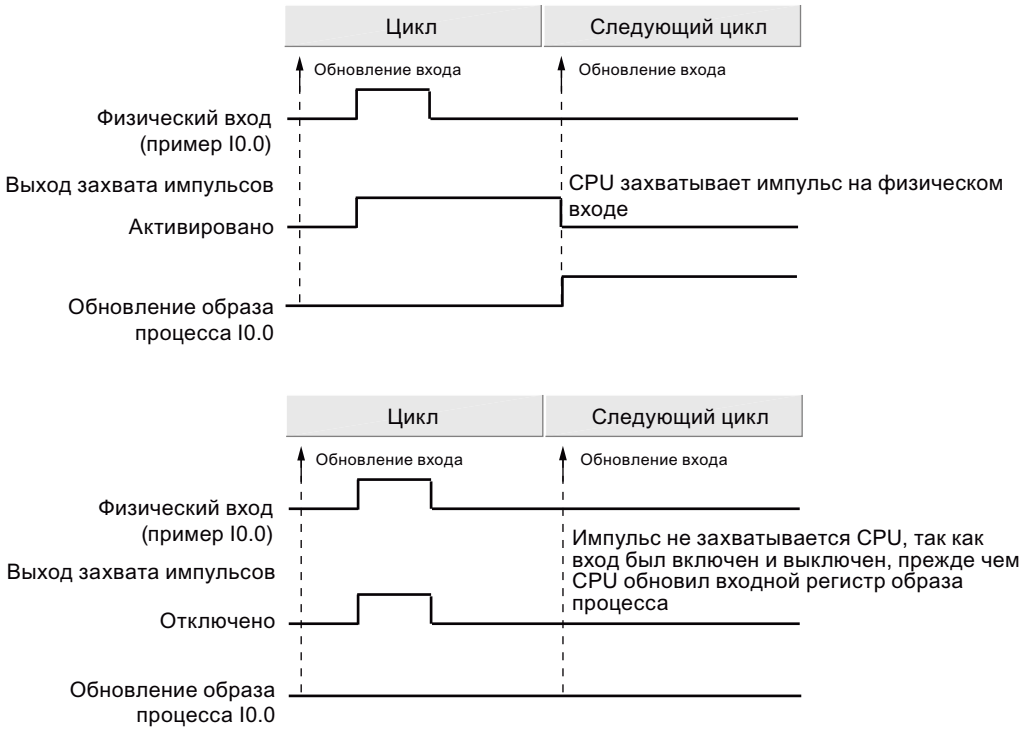
Тип высокоскоростного счётчика	Рекомендуемое время фильтра для входа
1 МГц	0,1 микросекунда
100 кГц	0,8 микросекунд
30 кГц	3,2 микросекунды

6.6.3 Захват импульсов

S7-1200 CPU предлагает захват импульсов для цифровых входов. Функция захвата импульсов позволяет захватывать импульсы высокого или низкого уровней, которые имеет такую короткую длительность, что они могут быть пропущены CPU, если цифровые входы читаются в начале цикла.

Когда для входа активирован захват импульсов, изменение состояния входа фиксируется и удерживается до следующего цикла обновления входа. Это гарантирует, что импульс, который активен лишь в течение короткого периода времени, будет захвачен и удержан, пока CPU читает входы.

Рисунок ниже показывает поведение CPU S7-1200 с и без активированного захвата импульсов:



Примечание

Поскольку функция захвата импульсов воздействует на вход после фильтра для входа, необходимо подобрать время фильтра для входа так, чтобы импульс не был отсеян фильтром.

Рисунок ниже показывает блок-схему цепи цифрового входа:

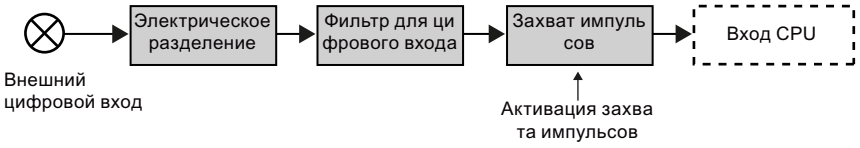
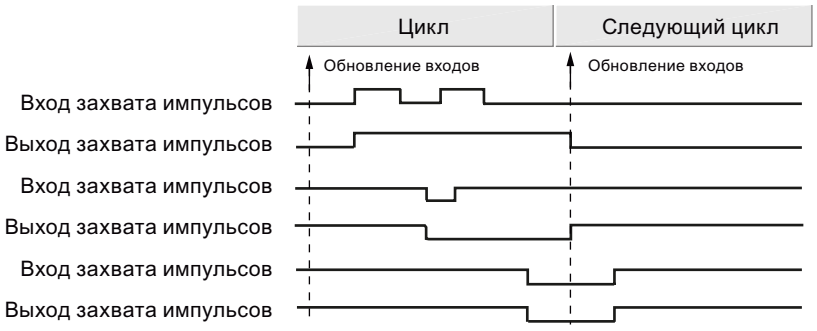


Рисунок ниже показывают реакцию активированной функции захвата импульсов на различные условия на входе. Если в определенном цикле присутствует больше чем один импульс, только первый импульс буден считан. Если в одном цикле присутствует несколько импульсов, то следует использовать события прерывания для переднего/заднего фронта:



6.7 Конфигурирование многоязыковой поддержки

Установки для многоязыковой поддержки позволяют назначить два языка проекта для каждого языка интерфейса S7-1200 веб-сервера (Страница 1023). Также можно не назначать язык проекта для языка интерфейса пользователя.

Что такое язык проекта?

Язык проекта это язык, на котором TIA Portal показывает пользовательские тексты проекта, напр., комментарии сегментов и комментарии блоков.

Выбор языка для отмеченного в дереве проекта осуществляется в TIA Portal с помощью команды меню **Инструменты > Языки проекта**.

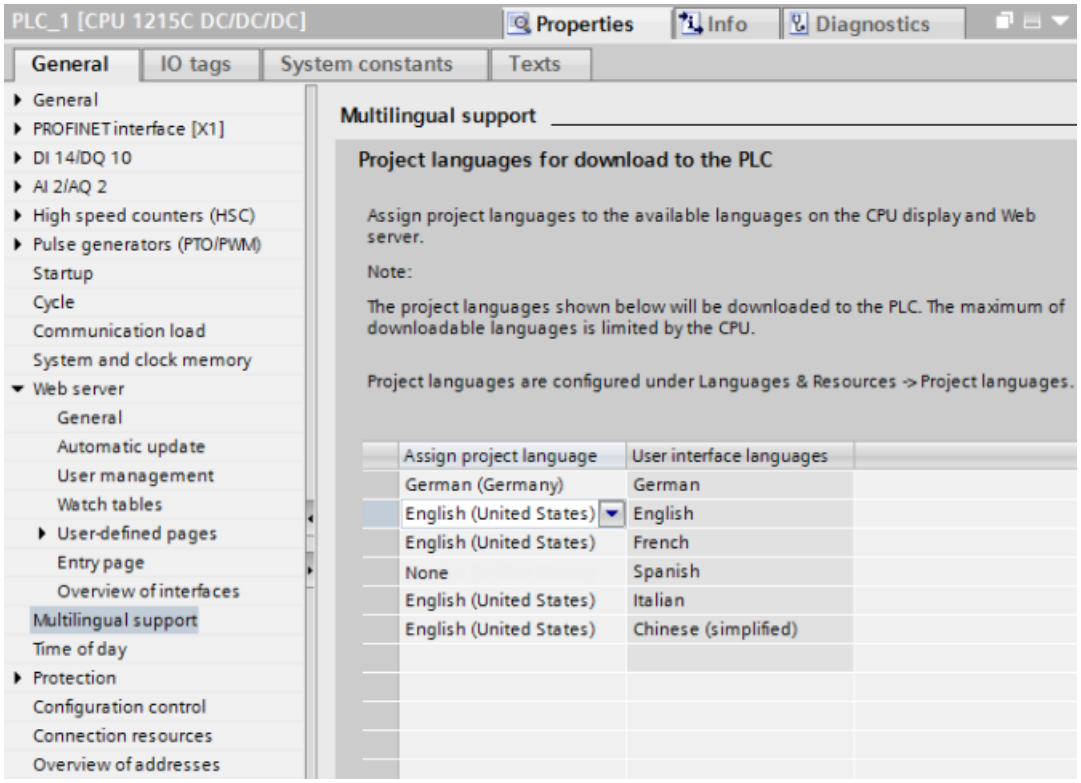
После через команду меню **Инструменты > Тексты проекта** можно конфигурировать тексты пользователя, напр., комментарии сегментов, на каждом из языков проекта. При изменении языка интерфейса для TIA Portal, комментарии сегментов, комментарии блоков и другие многоязычные тексты проекта отображаются на соответствующем языке проекта. Для установки языка интерфейса для TIA Portal на соответствующий язык проекта используется команда меню **Опции > Параметры**.

Для конфигурирования языков проекта и текстов проекта можно использовать и узел **Языки и ресурсы** дерева проекта.

Веб-сервер может использовать до двух языков проекта для STEP 7 для отображения информации из буфера диагностики.

Язык проекта и язык интерфейса веб-сервера

Веб-сервер поддерживает те же языки интерфейса, что и TIA Portal, но возможно максимум два языка проекта. Можно сконфигурировать веб-сервер таким образом, что, в зависимости от языка интерфейса веб-сервера, один из двух языков проекта будет использоваться для сообщений буфера диагностики. Эти параметры конфигурируются через свойства для "Многоязыковой поддержки" в конфигурации устройства CPU (комментарии сегментов, комментарии блоков и другие многоязычные тексты не доступны из веб-сервера).



В свойствах для Многоязыковой поддержки редактирование языков интерфейса на правой стороне невозможно. Речь идет о предопределенных языках, доступных как для TIA Portal, так и для интерфейса пользователя веб-сервера. Для конфигурирования параметра "Назначение языка проекта" можно выбрать один из двух доступных языков проекта или "Нет". Так как S7-1200 CPU поддерживает только два языка проекта, языком проекта не может быть тот же язык, что и язык интерфейса. Это правило распространяется на все поддерживаемые языки интерфейса.

В представленной ниже немецкой конфигурации Веб-сервер показывает записи в буфере диагностики (Страница 1045) (немецкий язык интерфейса пользователя веб-сервера) не отображаются тексты для событий буфера диагностики, если для интерфейса пользователя веб-сервера выбран испанский язык. Записи в буфере диагностики для всех других языков отображаются на английском языке.

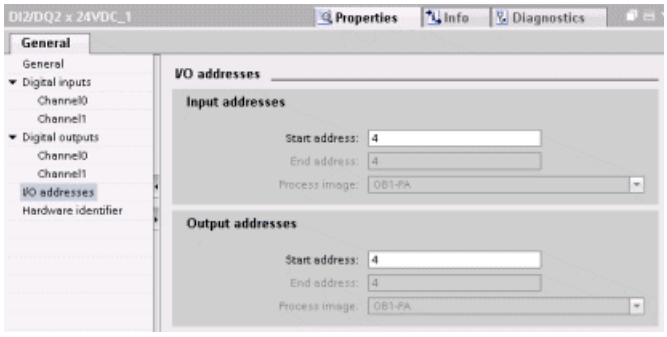
6.8 Конфигурирование параметров модулей

Для конфигурирования рабочих параметров модулей выбрать модуль в просмотре устройств и открыть вкладку "Свойства" в окне инспектора, чтобы установить параметры для модуля.

Конфигурирование сигнального модуля (SM) или сигнальной платы (SB)

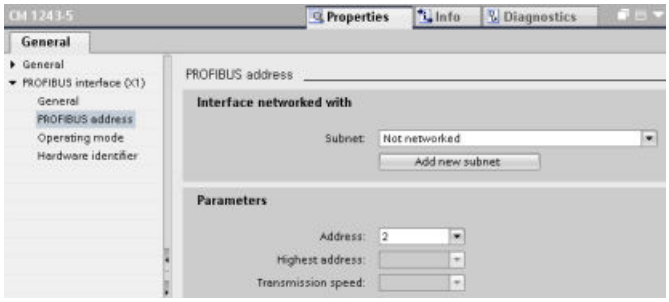
Конфигурация устройства для сигнальных модулей и сигнальных плат предлагает следующие возможности:

- Цифровые I/O: Можно сконфигурировать входы для обнаружения переднего или заднего фронта (с назначением события и аппаратного прерывания в каждом случае) или для "захвата импульса" (вход остается включенным после импульса) вплоть до следующего обновления образа процесса входов. Выходы могут быть деактивированы или могут подключаться подстановочное значение.
- Аналоговые I/O: Для отдельных входов конфигурируются параметры, напр., тип измерения (напряжение или ток), диапазон и сглаживание, а также разрешение для диагностики при выходе сигнала за верхний или нижний установленный предел. Аналоговые выходы предоставляют такие параметры, как тип выхода (по напряжению или току) и диагностика, например, короткое замыкание (для выходов по напряжению) или диагностика нарушения верхних или нижних предельных значений. Диапазоны аналоговых входов и аналоговых выходов в физических единицах не конфигурируются в диалоговом окне "Свойства". Для этого используется логика программы (см. "Обработка аналоговых значений (Страница 122)").
- Адреса I/O: Конфигурируется начальный адрес для входов и выходов модуля. Также можно назначить входы и выходы частичной области отображения процесса (PIP0, PIP1, PIP2, PIP3, PIP4) или выбрать для них автоматическое обновление или не использовать частичную область отображения процесса. Объяснение образов процесса и частичных областей отображения процесса см. "Выполнение программы пользователя" (Страница 83).



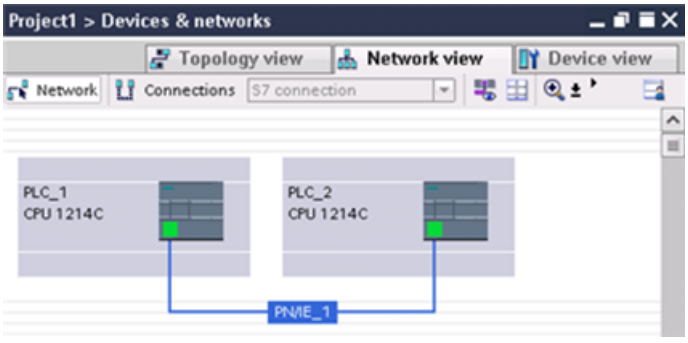
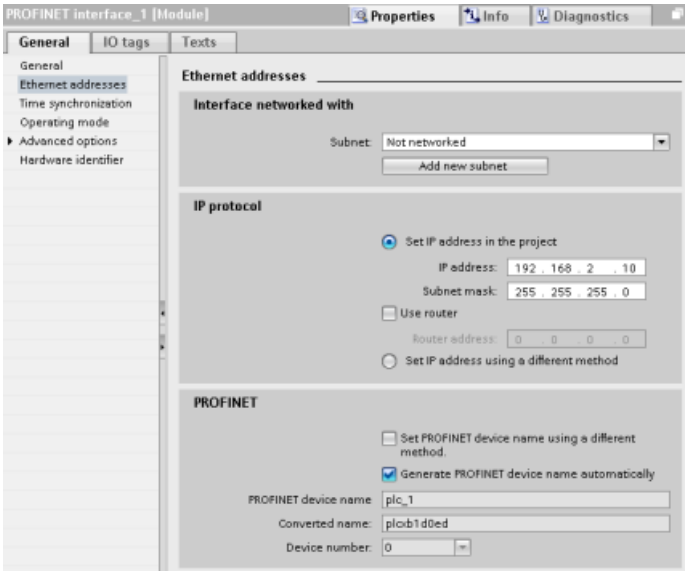
Конфигурирование коммуникационного интерфейса (CM, CP или CB)

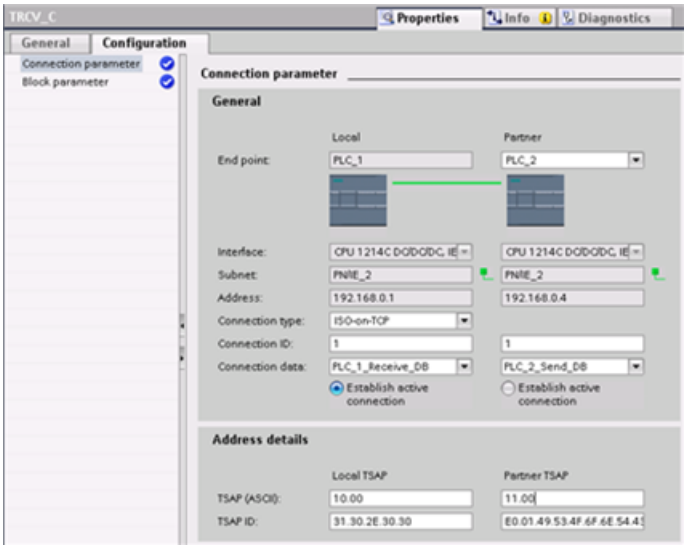
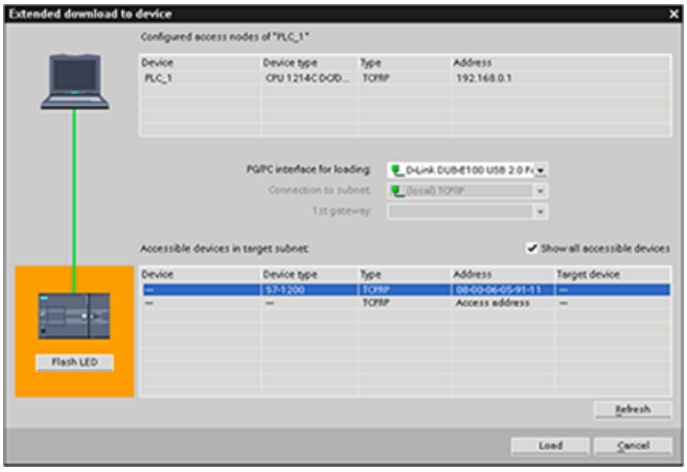
Параметры для сети конфигурируются по типу коммуникационного интерфейса.



6.9 Конфигурирование CPU для коммуникации

S7-1200 отвечает всем требованиям по коммуникации и передачи данных по сети, поддерживая не только простые, но и сложные сети. Кроме этого, S7-1200 предлагает инструменты для коммуникации с другими устройствами, напр., принтерами и весами, использующими собственные протоколы связи.

	На вкладке просмотра сетевых соединений конфигурации устройства можно создавать сетевые соединения между устройствами. После создания сетевого соединения открыть вкладку "Свойства" окна инспектора для конфигурирования параметров сети. Дополнительную информацию см. "Создание сетевого соединения" (Страница 811).
	Выбрать в диалоговом окне "Свойства" запись "Ethernet-адреса". STEP 7 отображает диалог для конфигурирования Ethernet адреса, в котором программному проекту присваивается IP-адрес модуля CPU, в который загружается проект. Примечание: У S7-1200 CPU нет предварительно сконфигурированного IP-адреса. Поэтому необходимо вручную назначить IP-адрес для CPU. Дополнительную информацию см. "Присвоение IP-адресов" (Страница 815).

	Для TCP, ISO-on-TCP и UDP Ethernet-протоколов, используют "свойства" инструкции (TSEND_C, TRCV_C или TCON), чтобы сконфигурировать соединения локального и партнерского CPU. На рисунке показаны "Свойства соединения" на вкладке "Соединение" для ISO-on-TCP соединения. Дополнительную информацию см. "Конфигурирование пути соединения между локальным и партнерским CPU" (Страница 812).
	После завершения конфигурирования загрузить проект в CPU. Все IP-адреса конфигурируются при загрузке проекта. Дополнительную информацию см. "Тестирование сети PROFINET" (Страница 823).

Примечание

Для подключения к CPU, сетевая карта (NIC) пользователя и ЦПУ должны относится к одному и тому же классу сети и той же самой подсети. Можно либо настроить свою сетевую карту согласно IP-адресу CPU по умолчанию, либо изменить IP-адрес CPU, чтобы он соответствовал классу сети и подсети сетевой карты пользователя.

Дополнительную информацию см. "Присвоение IP-адресов" (Страница 815).

6.10 Синхронизация времени

Целью синхронизации часов реального времени является создание эталонных часов, с которыми будут синхронизированы все другие локальные часы. Часы выполняются первичную синхронизацию с эталонными часами и после регулярно ее повторяют, чтобы не накапливать суммарную погрешность.

У S7-1200 и локальных базовых компонентов только модуль CPU и некоторые из CP модулей имеют часы реального времени, для которых может потребоваться синхронизация. Можно синхронизировать часы реального времени CPU с внешними эталонными часами. Внешние эталонные часы могут транслировать время через NTP сервер или CP модуль в локальной стойке S7-1200, который подключен к SCADA системе, в которой в свою очередь находятся эталонные часы.

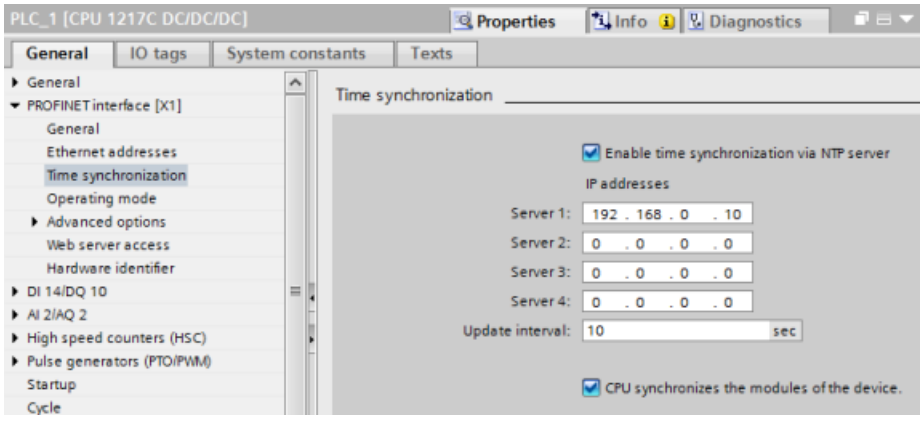
Дополнительную информацию о всех S7-1200 CP (https://support.industry.siemens.com/cs/us/en/ps), поддерживающих синхронизацию времени, можно найти на странице технической поддержки веб-сайта онлайн-поддержки для промышленности Siemens.

Установка часов реального времени

Предлагаются три различны способы для установки часов реального времени в S7-1200 CPU:

- Через NTP-сервер (Страница 826)
- Через STEP 7
- Через программу пользователя
- С помощью HMI панели

Для конфигурирования синхронизации времени модулей CP используются часы CPU, для чего установить флажок для "CPU синхронизирует модули устройства", как показано на рисунке ниже:



По умолчанию не активирована ни синхронизация времени через NTP-сервер, ни синхронизация часов CP с часами модуля CPU.

Синхронизация времени для часов модуля CPU и синхронизация времени для часов CP конфигурируются независимо друг от друга. Кроме этого, можно активировать синхронизацию времени для часов CP через CPU, если для установки часов CPU используется один из представленных выше методов.

При использовании NTP-сервера можно выбрать интервал обновления. По умолчанию в качестве интервала обновления для NTP-сервера выбраны 10 секунд.

При активации синхронизации времени в модуле, от STEP 7 поступает запрос на установку флажка "CPU синхронизирует модули устройства" в диалоговом окне "Синхронизация времени", если это не было сделано раньше. STEP 7 также выводит предупреждение, если несколько эталонных часов были сконфигурированы в качестве

источника для синхронизации времени (например, если синхронизация времени была активирована на нескольких CP модулях или на CPU и на одном модуле).

Примечание

Активация синхронизации времени на CP модуле приводит к тому, что этот CP модуль устанавливает часы CPU.

Если в диалоговом окне "Синхронизация времени" модуля CPU установлен флажок "CPU синхронизирует модули устройства", то CPU является эталонными часами. В этом случае CP модули синхронизируют часы CPU.

Примечание

Следует конфигурировать только один источник времени для CPU. Получение модулем CPU синхронизированного времени из нескольких источников (напр., от NTP-сервера и CP-модуля) может вызвать конфликты при обновлении времени. Синхронизация времени из нескольких источников может отрицательно сказаться на инструкциях и событиях, обрабатываемых на базе времени.

Основы программирования 7

7.1 Руководства по разработке решения автоматизации с устройством PLC

При разработке системы PLC можно выбирать из множества методов и критериев. Следующие общие руководящие принципы могут применяться ко многим проектам. Разумеется, необходимо придерживаться процедур, принятых в конкретной компании и учитывать собственный опыт.

Таблица 7-1 Руководства по разработке решения автоматизации с устройством PLC

Рекомендуемые шаги	Задачи
Структурирование процесса или системы	Следует разделить процесс или систему на части, не зависящие друг от друга. Эти части определяют границы между системами автоматизации и влияют на описание функциональных требований и распределение оборудования.
Описание функциональных требований	Сделать описания функций для каждой части процесса или системы, например, входы и выходы, функциональное описание процесса, состояния, которые должны быть достигнуты, прежде чем может начать реагировать исполнительное устройство (например, электромагнитный клапан, двигатель или привод), описание интерфейса оператора и всех интерфейсов с другими частями процесса или системы.
Разработка цепей безопасности	Определить устройства, которым для обеспечения безопасности может потребоваться схемно-реализованная логика. Следует помните, что устройства управления могут переходить в небезопасные рабочие состояния, что, в свою очередь, может привести к неожиданному поведению при запуске или изменениям в рабочих процессах оборудования. Там, где неожиданная или неправильная работа оборудования может привести к телесным повреждениям или существенному материальному ущербу, следует подумать об использовании электромеханических блокирующих устройств, работающих независимо от PLC, для предотвращения опасных рабочих состояний. Для разработки цепей безопасности действовать следующим образом: • Идентифицировать места, где возможно ненадлежащее или неожиданное функционирование исполнительных устройств, которое может вызвать опасное состояние. • Идентифицировать условия, при которых эксплуатация оборудования безопасна, и определить, как обнаружить эти условия независимо от PLC. • Определить, как PLC влияет на процесс, когда включается и снова отключается напряжение, а также при обнаружении ошибок. Использовать эту информацию только для проектирования нормальных и ожидаемых анормальных режимов работы. Из соображений безопасности не следует полагаться на этот сценарий "наилучшего случая". • Разработать цепи ручной или электромеханической защиты, с помощью которых опасные процессы блокируются независимо от PLC. • Обеспечить передачу соответствующей информации о состоянии от независимых цепей в PLC так, чтобы программа и интерфейсы операторов обладали необходимой информацией. • Определить другие требования техники безопасности для безопасного протекания процесса.

Рекомендуемые шаги	Задачи
Планирование безопасности системы	Определить, какая степень защиты (Страница 203) необходима для доступа к процессу. Для защиты модулей CPU и программных блоков от несанкционированного доступа можно использовать пароль.
Определение станций оператора	Создать следующие схемы станций оператора на основе описаний в функциональных требованиях: • Расположение всех станций оператора относительно процесса / установки • Механическое расположение устройств на станции оператора, напр., дисплея, коммутатора и световых индикаторов. • Электрические схемы с соответствующими входами и выходами PLC и сигнальных модулей
Разработка конфигурационных схем	На основе описаний функциональных требований, разработать конфигурационные схемы для системы автоматизации: • Расположение всех PLC относительно процесса / установки • Механическое расположение всех PLC и модулей I/O (включая шкафы и т.п.) • Электрические схемы для всех PLC и модулей I/O, включая номер устройства, коммуникационные адреса и адреса для входов и выходов
Создание списка символьных имен	Создать список символьных имен для абсолютных адресов. Указать не только физические входы и выходы, но также и все другие элементы (например, имена переменных (тегов)), которые используются в программе.

7.2 Структурирование программы пользователя

При создании программы пользователя для решения задачи автоматизации инструкции для программы вставляются в блоки кода:

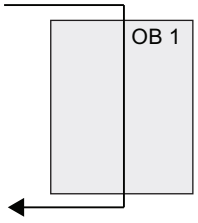
- Организационный блок (OB) реагирует на определенное событие в CPU и может прервать выполнение программы пользователя. Стандартный блок для циклического выполнения программы пользователя (OB 1) предоставляет основную структуру для программы пользователя. Вставка других OB в программу прерывает выполнение OB 1 Другие OB выполняют специфические функции, например, для запуска, обработки прерываний и ошибок или для выполнения конкретного программного кода через определенные интервалы времени.
- Функциональный блок (FB) – это подпрограмма, которая выполняется при вызове из другого блока кода (OB, FB или FC). Вызывающий блок передает параметры в FB и определяет определенный блок данных (DB), который сохраняет данные для этого вызова или этого экземпляра FB. Изменение DB экземпляра позволяет общему FB управлять работой группы оборудования. Например, один FB может управлять несколькими насосами или клапанами с различными DB экземпляра, содержащими специальные рабочие параметры для каждого насоса или клапана.
- Функция (FC) – это подпрограмма, которая выполняется при вызове из другого блока кода (OB, FB или FC). У FC нет связанного DB экземпляра. Вызывающий блок передает параметры в FC. Выходные значения FC должны быть записаны в ячейку памяти или в глобальный DB.

Выбор типа структуры для программы пользователя

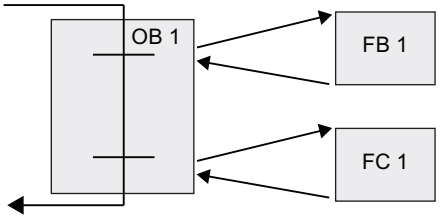
В зависимости от требований приложения, можно выбрать для программы пользователя линейную или модульную структуру:

- Линейная программа выполняет все команды для задач автоматизации последовательно друг за другом. Обычно в линейной программе все инструкции находятся в OB цикла (OB 1).
- Модульная программа вызывает специальные блоки кода, которые выполняют конкретные задачи. Для создания модульной структуры программы, сложная задача автоматизации делится на небольшие подзадачи, соответствующие технологическим функциям процесса. Каждый блок кода содержит сегмент программы для соответствующей подзадачи. Для структурирования программы используется вызов один блок кода из другого блока.

Линейная структура:



Модульная структура:



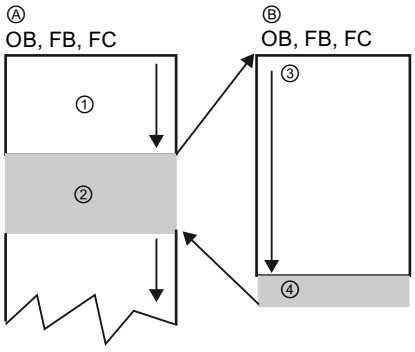
Разработку и реализацию программы пользователя можно упростить, создавая общие блоки кода, которые могут быть многократно использованы в программе пользователя. Работа с общими блоками кода имеет ряд преимуществ:

- Можно создавать повторно используемые блоки кода для стандартных задач, например, для управления насосом или двигателем. Кроме этого, можно хранить эти общие блоки кода в библиотеке, которая может быть использована для различных приложений или решений.
- Модульное структурирование программы пользователя на отдельные компоненты, соответствующие функциональным задачам, делает программу более наглядной и более простой в работе. Модульные компоненты не только помогают стандартизовать разработку программы, но также ускоряют и упрощают адаптацию или модификацию программного кода.
- Создание модульных компонентов упрощает отладку программы. Если вся программа разделена на последовательность модульных сегментов, то можно тестировать функции каждого блока кода непосредственно во время разработки.
- Создание модульных компонентов, относящихся к конкретным технологическим функциям, упрощает и сокращает ввод в эксплуатацию всего приложения.

7.3 Использование блоков для структурирования программы

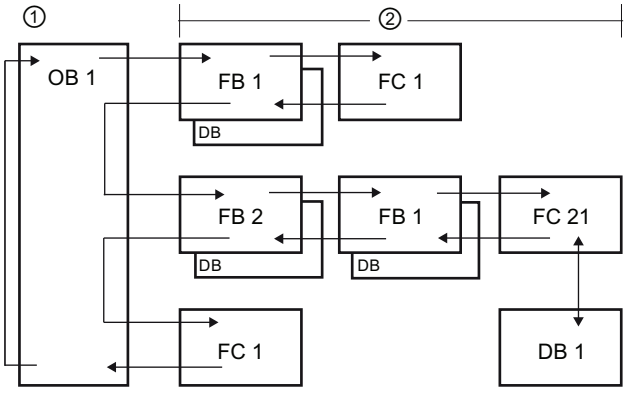
Модульные блоки кода создаются путем разработки FB и FC для выполнения общих задач. Затем программа структурируется, при этом другие блоки кода вызывают эти многоразовые модули. Вызывающий блок передают параметры конкретного устройства в вызываемый блок.

Если блок кода вызывается другим блоком кода, то CPU выполняет программный код в вызванном блоке. После того как вызванный блок обработан, CPU возобновляет выполнение вызывающего блока. Обработка продолжается выполнением инструкции, следующей за вызовом блока.



A Вызывающий блок
B Вызываемый (или прерывающий) блок
① Выполнение программы
② Инструкция или событие, запускающее выполнение другого блока
③ Выполнение программы
④ Конец блока (возврат в вызывающий блок)

Вызовы блоков могут быть вложены друг в друга, делая структуру еще более модульной. В следующем примере используется 3 уровня вложенности: OB программного цикла плюс 3 уровня с вызовами блоков кода.



① Начало цикла
② Глубина вложенности

Примечание: Максимальная глубина вложенности составляет шесть уровней. Программы обеспечения безопасности используют два уровня вложенности. Таким образом, программа пользователя имеет в программах обеспечения безопасности четыре уровня вложенности.

7.3.1 Организационный блок (OB)

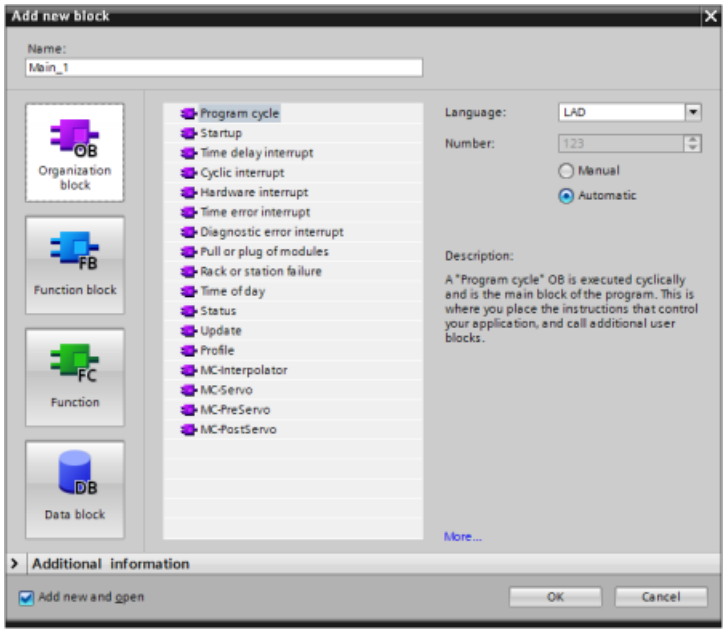
Организационные блоки служат для структурирования программы. Они формируют интерфейс между операционной системой и программой пользователя. OB управляются событиями. Событие, например, диагностическое прерывание или интервал времени, инициирует исполнение OB модулем CPU. Некоторые OB имеют предопределенные стартовые события и поведение.

OB программного цикла содержит главную программу. Можно включить в программу пользователя боле одного OB программного цикла. В рабочем состоянии RUN, OB программного цикла выполняются с наименьшим уровнем приоритета и могут быть прерваны всеми другими типами событий. Пусковой OB не прерывает OB программного цикла, так как CPU выполняет пусковой OB до перехода в режим RUN.

После обработки OB программного цикла, CPU немедленно исполняет этот OB снова. Такая циклическая обработка является "штатным" режимом выполнения для программируемых логических контроллеров. Во многих приложениях вся программа пользователя содержится в одном единственном OB программного цикла.

Могут создавать и другие OB для выполнения определенных функций, например, для обработки прерываний и ошибок или для выполнения конкретного программного кода через определенные интервалы времени. Такие OB прерывают выполнение OB программного цикла.

Для создания новых OB для программы пользователя используется диалоговое окно "Добавление нового блока".



Обработка прерываний всегда управляется событиями. Когда такое событие происходит, CPU прерывает выполнение программы пользователя и вызывает OB для обработки этого события. После выполнения прерывающего OB, CPU возобновляет обработку программы пользователя в месте прерывания.

CPU определяет порядок обработки прерывающих событий на основе приоритетов. Несколько прерывающих событий могут быть объединены в классы приоритетов. Дополнительную информацию можно найти в разделах, рассматривающих тематику организационных блоков (Страница 91) и выполнения программы пользователя (Страница 83).

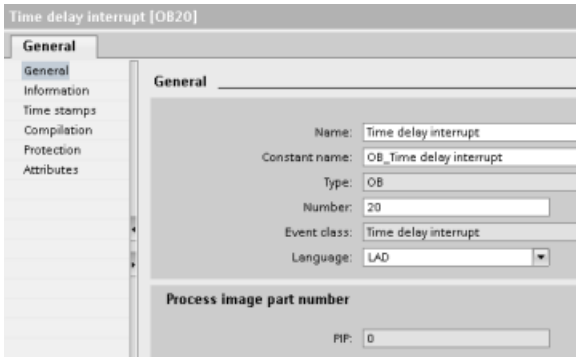
Создание дополнительных OB

Можно создать несколько OB для программы пользователя. Это также касается событий OB программного цикла и пусковых OB. В диалоговом окне "Добавить новый блок" можно создать OB и ввести для него имя.

Если создается несколько OB программного цикла для программы пользователя, то CPU выполняет каждый OB программного цикла в числовой последовательности. При этом выполнение начинается с OB программного цикла с наименьшим номером (напр., OB 1). Пример: После завершения первого OB программного цикла (напр., OB 1), CPU выполняет OB программного цикла в порядке возрастания номеров.

Конфигурирование свойств OB

Можно изменять свойства OB, например, сконфигурировать номер OB или язык программирования.



Примечание

Следует помнить, что можно присвоить OB номер частичной области отображения процесса, напр., TPA0, TPA1, TPA2, TPA3 или TPA4. Если ввести для частичной области отображения процесса номер, то CPU создаст частичную область отображения процесса. Объяснение частичных областей отображения процесса см. "Выполнение программы пользователя (Страница 83)".

7.3.2 Функция (FC)

Функция (FC) – это блок кода, который обычно выполняет определенную операцию с набором входных значений. FC сохраняет результаты этой операции в определенных ячейках памяти. Например, можно использовать FC, чтобы выполнять стандартные и допускающие повторное использование операции (такие, как математические вычисления) или технологические функции (такие, как отдельные процессы управление с использованием битовых логических операций). FC может многократно вызываться в различных местах внутри программы. Такое повторное использование упрощает программирование часто повторяющихся задач.

У FC нет связанного DB экземпляра. FC использует локальный стек данных для временных данных, необходимых в процессе вычисления операций. Временные данные

не сохраняются. Для сохранения данных на постоянной основе, присвоить выходное значение ячейке глобальной памяти, напр., M памяти или к глобальному DB.

7.3.3 Функциональный блок (FB)

Функциональный блок (FB) - это блок кода, который использует блок данных экземпляра для своих параметров и статических данных. FB имеют память переменных, которая находится в блоке данных (DB) или в DB экземпляра. DB экземпляра предоставляет блок памяти, который назначен этому экземпляру (или этому вызову) FB, и сохраняет данные после выполнения FB. Можно назначать различным вызовам FB различные DB экземпляра. DB экземпляра позволяет использовать общий FB для управления несколькими устройствами. Можно структурировать программу, при этом один блок вызывает один FB и один DB экземпляра. Затем CPU исполняет программный код в этом FB и сохраняет параметры блока и статические локальные данные в DB экземпляра. Когда исполнение FB заканчивается, CPU продолжает обработку с блока кода, который вызвал FB. DB экземпляра сохраняет значения для этого экземпляра FB. Эти значения доступны для последующих вызовов функционального блока в том же самом или в других циклах.

Повторно используемые блоки кода с выделенной памятью

Обычно FB используется для управления последовательностью действий или устройствами, которые не заканчивают свою работу в течение одного цикла. Для сохранения рабочих параметров так, чтобы к ним можно было быстро обращаться между двумя циклами, каждый FB в программе пользователя имеет один или несколько DB экземпляра. При вызове FB одновременно указывается DB экземпляра, содержащий параметры блока и статические локальные данные для этого вызова или этого "экземпляра" FB. DB экземпляра сохраняет эти значения по окончании выполнения FB.

Если FB разрабатывается для общих задач управления, то он может использоваться для нескольких устройств, при этом различные DB экземпляра выбираются для различных вызовов FB.

FB сохраняет входные, выходные и IO-параметры, а также статические параметры, в один DB экземпляра.

Кроме этого, можно изменять и загружать интерфейс функционального блока в рабочем состоянии RUN (Страница 1416).

Назначение стартового значения в DB экземпляра

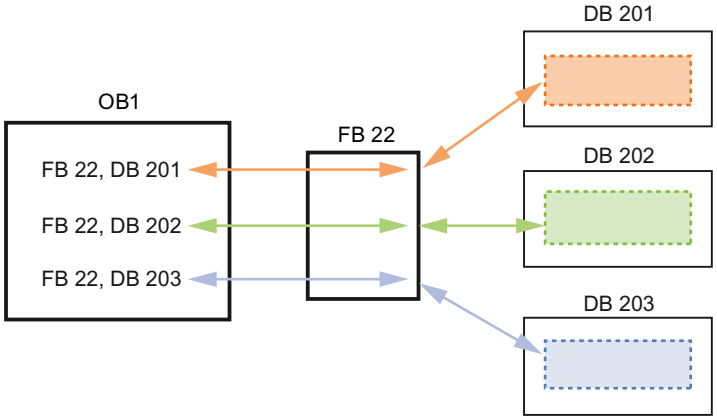
DB экземпляр хранит как значение по умолчанию, так и стартовое значение для каждого параметра. Стартовое значение указывает значение, которое должно использоваться при выполнении FB. Стартовое значение затем может быть изменено во время обработки программы пользователя.

На интерфейсе FB есть столбец для значения по умолчанию, который позволяет ввести новое стартовое значение для параметра при разработке программного кода. Это значение по умолчанию в FB затем передается в стартовое значение соответствующего DB экземпляра Если новое стартовое значение не присваивается параметру на

интерфейсе FB, то значение по умолчанию из DB экземпляра копируется в стартовое значение.

Использование одного FB с несколькими DB

На следующем рисунке показан OB, который трижды вызывает FB, используя разные блоки данных для каждого вызова. Эта структура позволяет одному и тому же FB управлять несколькими одинаковыми устройствами, например, двигателями, назначая каждому вызову устройства свой блок данных экземпляра. Каждый DB экземпляра хранит данные (например, скорость, время запуска и общее время работы) для отдельного устройства.



В этом примере FB 22 управляет тремя различными устройствами, DB 201 хранит эксплуатационные данные для первого устройства, DB 202 хранит эксплуатационные данные для второго устройства, и DB 203 хранит эксплуатационные данные для третьего устройства.

7.3.4 Блок данных (DB)

В программе пользователя можно создавать блоки данных (DB) для сохранения данных для блоков кода. Все программные блоки в программе пользователя могут обращаться к данным в глобальном DB, но DB экземпляра хранит данные для конкретного функционального блока (FB).

Сохраненные данные в DB после выполнения соответствующего блока кода не удаляются. Существуют два типа блоков данных:

- Глобальный DB хранит данные для блоков кода в программе. Каждый OB, FB или FC может обратиться к данным в глобальном DB.
- DB экземпляр хранит данные для определенного FB. Структура данных в DB экземпляра соответствует параметрам (Input, Output и InOut) и статическим данным FB. (Временная память FB не сохраняется в DB экземпляра.)

Примечание

Хотя DB экземпляра содержит данные для конкретного FB, любой блок кода может получить доступ к данным DB экземпляра.

Можно изменять и загружать блоки данных в рабочем состоянии RUN (Страница 1416).

Блоки данных только для чтения

Можно сконфигурировать DB так, что он будет доступен только для чтения:

1. Кликнуть правой кнопкой мыши в дереве проекта по DB и выбрать в контекстном меню "Свойства".
2. Выбрать в диалоговом окне "Свойства" элемент "Атрибуты".
3. Выбрать опцию "Защита от записи для блока данных в устройстве" и подтвердить нажатием OK.

Оптимизированные и стандартные блоки данных

Можно сконфигурировать блок данных в качестве стандартного или оптимизированного блока данных. Стандартный DB совместим с инструментами программирования STEP 7 Classic и классическими CPU S7-300 и S7-400. Блоки данных с оптимизированным доступом не имеют фиксированной структуры. В объявлении элемента данных содержится только одно символьное имя, в блоке отсутствует фиксированный адрес. CPU сохраняет элементы автоматически в доступной области памяти блока так, чтобы не было никаких пропусков в памяти. Это делается для оптимального использования объема памяти.

Для установки оптимизированного доступа для блока данных, выполнить следующие шаги:

1. Раскрыть папку программных блоков в дереве проекта STEP 7.
2. Кликнуть правой кнопкой мыши по блоку данных и выбрать команду "Свойства" из контекстного меню.
3. Выбрать в атрибутах опцию "Оптимизированный доступ к блоку".

Обратить внимание на то, что оптимизированный доступ к блоку выбирается по умолчанию для новых блоков данных. Если отключить "Оптимизированный доступ к блоку", то для блока используется стандартный доступ.

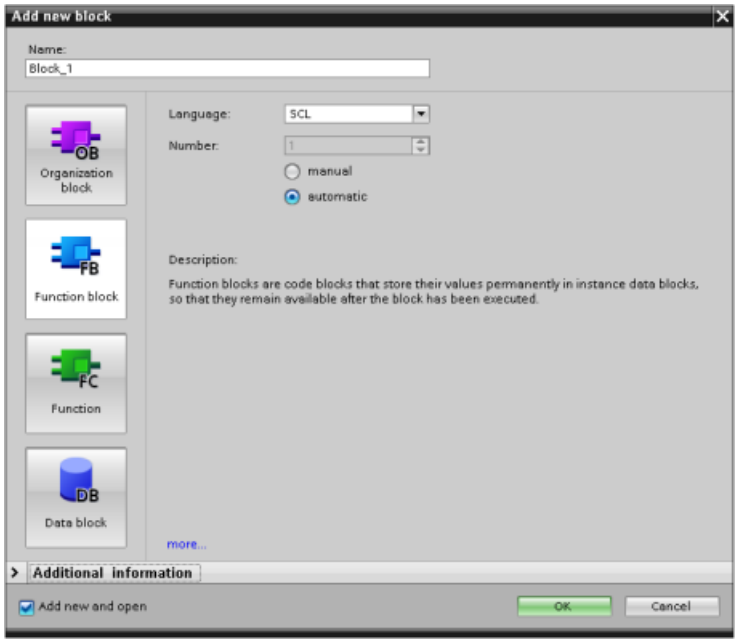
Примечание

Тип доступа к блоку для FB и его DB экземпляра

Если для FB выбрана настройка "Оптимизированный доступ к блоку", тогда настройкой для DB экземпляра этого FB также должен быть "Оптимизированный доступ к блоку". Точно также, если "Оптимизированный доступ к блоку" не был выбран для FB, то есть доступ к FB осуществляется стандартным образом, то и для DB экземпляра должен быть установлен стандартный доступ, а не опция "Оптимизированный доступ к блоку".

Если типы доступа несовместимы, то при выполнении FB через устройство HMI могут быть потерянные внесенные в IN/OUT значения параметров FB изменения.

7.3.5 Создание повторно используемых блоков кода



Для создания OB, FB, FC и глобальных DB, открыть в дереве проекта диалоговое окно "Добавить новый блок" в разделе "Программные блоки".

При создании блока кода следует выбрать язык программирования для блока. Для DB язык не выбирается, так он лишь хранит данные.

При установке флажка "Добавить и открыть" (установка по умолчанию) блок кода открывается в проектном представлении.

Объекты, которые будут использованы повторно, могут быть сохранены в библиотеку. У каждого проекта есть связанная с ним библиотека. В дополнение к библиотеке проекта может создаваться любое число глобальных библиотек для использования в нескольких проектах. Так как библиотеки совместимы между собой, можно копировать или перемещать элементы между библиотеками.

Библиотеками используются, например, для создания шаблонов блоков, которые копируются в библиотеку проекта для дальнейшей обработки. В заключении, блоки копируются из библиотеки проекта в глобальную библиотеку. Глобальная библиотека доступна для других коллег, работающих над проектом. Они используют блоки и могут адаптировать их к своим индивидуальным требованиям, в случае необходимости.

Дополнительную информацию о работе с библиотеками можно найти системе интерактивной помощи STEP 7.

7.3.6 Передача параметров в блоки

У функциональных блоков (FB) и функций (FC) есть три различных типа интерфейса:

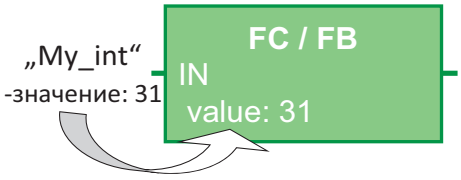
- IN
- IN/OUT
- OUT

FB и FC получают параметры через интерфейсы типа IN и IN/OUT. Блоки обрабатывают параметры и возвращают значения вызывающему блоку через интерфейсы типа IN/OUT и OUT.

Программа пользователя передает параметры, используя один из двух методов:

Передача по значению

Если программа пользователя передает параметр в функцию как "Вызов по значению", то программа пользователя копирует фактическое значение параметра во входной параметр блока для интерфейса типа IN. Эта операция требует дополнительной памяти для скопированного значения.



Когда пользовательская программа вызывает блок, она копирует значения.

Передача по ссылке

Если программа пользователя передает параметр в функцию как "Вызов по ссылке", то программа пользователя ссылается на адрес фактического параметра для интерфейса типа IN/OUT и не копирует значение. Эта операция не требует дополнительной памяти.



Когда пользовательская программа вызывает блок, она ссылается на адрес фактических параметров.

Примечание

Как правило, тип интерфейса IN/OUT используют для структурированных переменных (тегов) (например, ARRAY, STRUCT и STRING) во избежание излишнего увеличения требуемой памяти данных.

Оптимизация блока и передача параметров

Программа пользователя передает параметры FC как "вызов по значению" для простых типов данных (например, INT, DINT и REAL). Она передает сложные типы данных (например, STRUCT, ARRAY и STRING) как "вызов по ссылке".

Программа пользователя обычно передает параметры FB в относящемся к FB блоке данных экземпляра (DB):

- Программа пользователя передает простые типы данных (например, INT, DINT и REAL) как "вызов по значению", копируя параметры в/из DB экземпляра.
- Программа пользователя копирует сложные типы данных (например, STRUCT, ARRAY и STRING) в и из DB экземпляра для параметров типа IN и OUT.
- Программа пользователя передает сложные типы данных как "вызов по ссылке" для интерфейса типа IN/OUT.

DB могут быть созданы как "оптимизированные" или как "стандартные" (неоптимизированные). Оптимизированные блоки данных более компактны, чем неоптимизированные блоки данных. Кроме того, порядок элементов данных в DB отличается у оптимизированных и неоптимизированных DB. В разделе "Оптимизированные блоки" в Руководстве по S7 программированию для S7-1200/1500, STEP 7 (TIA Portal), 03/2014 (http://support.automation.siemens.com/WW/view/en/81318674) можно найти сведения об оптимизированных блоках.

FB и FC создаются для обработки оптимизированных или неоптимизированных данных. Можно установить флажок "Оптимизированный доступ к блоку" как один из атрибутов для блока. Программа пользователя оптимизирует программные блоки по умолчанию, и программные блоки ожидают данные, передаваемые в блок в оптимизированном формате.

Когда программа пользователя передает сложный параметр (например, STRUCT) в функцию, система проверяет установку оптимизации для блока данных со структурой и установку оптимизации программного блока. Если оптимизируется и блок данных и функция, то программа пользователя передает STRUCT как "вызов по ссылке". Также происходит, если ни для блока данных, ни для функции оптимизация не выполняется.

Однако, если функция и блок данных имеют разные установки оптимизации (т.е. один блок оптимизирован, а другой - нет), то STRUCT должен быть преобразован в формат, ожидаемый функцией. Пример: Например, если блок данных не оптимизирован, а функция оптимизирована, то STRUCT в блоке данных должен быть преобразован в оптимизированный формат, чтобы функция могла обрабатывать STRUCT. Система выполняет это преобразование, делая "копию" STRUCT и преобразовывая его в оптимизированный формат, который ожидает функция.

Таким образом, когда программа пользователя передает сложный тип данных (например, STRUCT) в функцию как параметр IN/OUT, функция ожидает, что программа пользователя передаст STRUCT как "вызов по ссылке":

- Если выбирается оптимизация или ее отсутствие как для блока данных с STRUCT, так и для функции, то программа пользователя передает данные как "вызов по ссылке".
- Если для блока данных и функцию конфигурируются разные настройки оптимизации (один блок оптимизирован, а другой нет), система должна сделать копию STRUCT до передачи его в функцию. Поскольку система должна сделать эту копию структуры, она фактически преобразовывает "вызов по ссылке" в "вызов по значению".

Влияние настройки оптимизации на программы пользователя

Копирование параметра может вызвать проблему в программе пользователя, если HMI или OB прерывания изменяет элементы структуры. Пример: Например, есть параметр IN/OUT функции (обычно передаваемый как "вызов по ссылке"), но параметры настройки оптимизации блока данных и функции отличаются:

1. Когда программа пользователя готова вызвать функцию, система должна сделать "копию" структуры, чтобы согласовать формат данных под функцию.
2. Программа пользователя вызывает функцию со ссылкой на "копию" структуры.
3. OB прерывания вызывается при выполнении функция и изменяет значение в исходной структуре.
4. Функция выполняется и, так как структура является параметром IN/OUT, система копирует значения назад в исходную структуру в исходном формате.

Т.к. была создана копия структуры для изменения формата, записанные OB прерывания данные теряются. То же может произойти при записи значения с устройством HMI. Устройство HMI может прервать программу пользователя и записать значение таким же образом, как OB прерывания.

Есть несколько способов избежать этих проблем:

- Лучшее решение этой проблемы состоит в том, чтобы выбрать идентичными настройки оптимизации программного блока и блока данных при использовании сложных типов данных (например, STRUCT). Это гарантирует, что программа пользователя всегда передает параметры как "вызов по ссылке".
- Другое решение состоит в том, что OB прерывания или устройство HMI не изменяют напрямую элемент в структуре. OB или устройство HMI могут изменить другую переменную, а после эта переменная может быть скопирована в структуру в определенной точке программы пользователя.

7.4 Целостность данных

CPU поддерживает целостность данных для всех элементарных типов данных (например, Word или DWord) и всех определяемых системой структур (например, IEC_TIMERS или DTL). Процесс чтения или записи такого значения не может быть прерван. (Например, CPU защищает доступ к значению типа DWord, пока все четыре байта DWord не будут прочитаны или записаны.) Чтобы гарантировать невозможность одновременной записи в одну и ту же ячейку памяти со стороны OB программного цикла и OB прерываний, CPU не выполняет OB прерываний, пока операция чтения или записи в OB программного цикла не будет завершена.

Если в программе пользователя несколько значений в памяти используются OB программного цикла и OB прерываний совместно, то программа пользователя должна обеспечить целостность при изменении или считывании этих значений. Можно использовать инструкции DIS_AIRT (выключить обработку прерываний) и EN_AIRT (включить обработку прерываний), чтобы защитить доступ к совместно используемым значениям в OB программного цикла.

- Вставить в кодовый блок инструкцию DIS_AIRT, чтобы не допустить исполнения OB прерываний во время операции чтения или записи.
- Вставите инструкции для чтения или записи значений, которые может изменить OB прерываний.
- Вставить инструкцию EN_AIRT в конце этой последовательности, чтобы отменить инструкцию DIS_AIRT и разрешить исполнение OB прерываний.

Коммуникационный запрос от устройства человеко-машинного интерфейса или другого CPU также может прервать исполнение OB программного цикла. Коммуникационные запросы также могут привести к проблемам с целостностью данных. CPU обеспечивает целостность элементарных типов данных при их считывании и записи инструкциями в программе пользователя. Так как программа пользователя регулярно прерывается коммуникационными запросами, то нельзя гарантировать, что несколько значений в CPU будут все обновлены одновременно устройством HMI. Например, значения, отображаемые на экране устройства человеко-машинного интерфейса, могут происходить из различных циклов CPU.

Инструкции PtP (точка-точка), инструкции PROFINET (напр., TSEND_C и TRCV_C), инструкции PROFINET для распределенной периферии (Страница 386) и инструкции PROFIBUS для распределенной периферии (Страница 386) передают буферные области данных, которые могут быть прерваны. Следует обеспечить целостность данных для буферных областей, избегая операций чтения или записи для этих буферных областей как в OB программного цикла, так и в OB прерываний. Если все же необходимо изменить значения в буфере для этих инструкций в OB прерываний, воспользоваться инструкцией DIS_AIRT, чтобы отложить прерывание (через OB прерываний или коммуникационное прерывание от устройства человеко-машинного интерфейса или другого CPU), пока не будет выполнена инструкция EN_AIRT.

Примечание

Использование инструкции DIS_AIRT задерживает обработку OB прерываний до тех пор, пока не будет выполнена инструкция EN_AIRT, и влияет на латентность прерывания (интервал времени от возникновения события до исполнения OB прерываний) в программе пользователя.

7.5 Язык программирования

STEP 7 предлагает следующие стандартные языки программирования для S7-1200:

- LAD (релейно-контактные схемы) - это графический язык программирования. Для отображения используются принципиальные схемы (Страница 192).
- FBD (функциональная блок-схема) - это язык программирования на основе графических логических символов Булевой алгебры (Страница 193).
- SCL (структурированный язык управления) - это язык программирования высокого уровня на основе текстов (Страница 194).

При создании блока кода следует выбрать язык программирования для блока.

Программа пользователя может работать с блоками кода, созданными с помощью одного из этих языков программирования.

7.5.1 Релейно–контактная схема (LAD)

Элементы принципиальной схемы, такие как нормально замкнутые, нормально разомкнутые контакты и катушки соединяются в сегменты.



Для создания логики сложных операций можно вставлять ветвления для параллельных цепей. Параллельные ветвления открыты вниз или соединены непосредственно с шиной питания. Разработчик завершает ветвления вверх.

LAD предлагает блочные инструкции для множества функций, напр., арифметических операций, таймеров, счетчиков и функций перемещения.

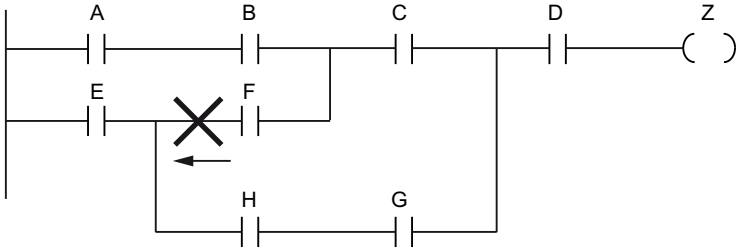
STEP 7 не ограничивает макс. число инструкций (строки и столбцы) в сегменте LAD.

Примечание

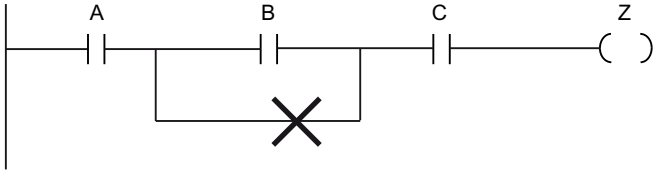
Каждый сегмент LAD должен завершаться катушкой или блоком.

При создании сегмента LAD необходимо соблюдать следующие правила:

- Нельзя создавать ветвления, которые могут привести к реверсированию потока сигналов.

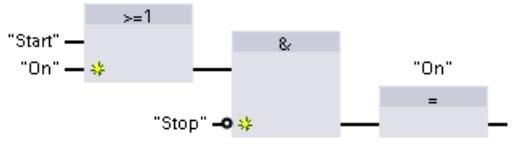


- Нельзя создавать ветвления, которые могут привести к короткому замыканию.



7.5.2 Функциональная блок-схема (FBD)

Как и LAD, FBD - это тоже графический язык программирования. Представление логики здесь основано на графических символах, используемых в булевой алгебре.



Для создания логических связей между сложными операциями, вставить параллельные ветвления между блоками.

Арифметические и другие сложные функции могут быть представлены непосредственно в соединении с логическими блоками.

STEP 7 не ограничивает макс. число инструкций (строки и столбцы) в сегменте FBD.

7.5.3 SCL

Структурированный язык управления SCL (Structured Control Language, SCL) - это язык программирования высокого уровня для CPU SIMATIC S7 на базеPASCAL. SCL поддерживает блочную структуру STEP 7 (Страница 182). Проект может включать программные блоки, созданные на любом из этих трех языков программирования: SCL, LAD и FBD.

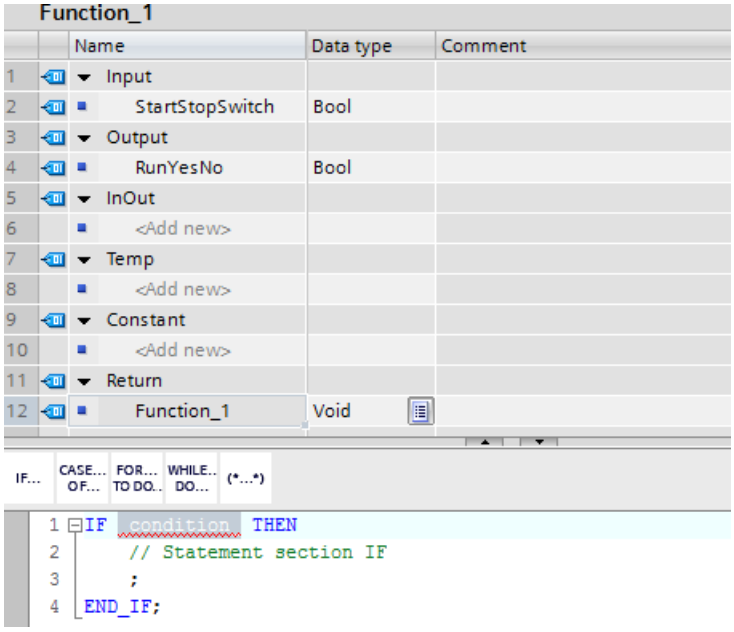
Инструкции SCL используют типовые операторы программирования, такие как присваивание (: =), математические функции (+ для сложения, - для вычитания, * для умножения, и / для деления). Также SCL использует стандартные операции PASCAL для программного управления, напр., IF-THEN-ELSE, CASE, REPEAT-UNTIL, GOTO иRETURN. Для синтаксических элементов языка программирования SCL можно использовать любые ссылки PASCAL. Многие из других инструкций для SCL, напр., таймеры и счетчики, соответствуют инструкциям LAD и FBD. Для получения более подробной информации о специфических инструкциях см. описания этих инструкций в главах Базовые инструкции (Страница 219) и Расширенные инструкции (Страница 343).

7.5.3.1 Редактор текстов программы SCL

При создании любых типов блоков (OB, FB или FC) можно указать, что блок должен использовать язык программирования SCL. STEP 7 содержит редактор текстов программы SCL со следующими элементами:

- Интерфейсный раздел для определения параметров блока кода
- Раздел для программного кода
- Дерево инструкций, которое содержит инструкции SCL, поддерживаемые CPU

Код SCL для инструкции вводится напрямую в разделе кода. Редактор содержит кнопки для общих конструкций кода и комментариев. Для создания более сложных инструкций просто перетащить инструкции SCL мышью из дерева инструкций в программу. Также можно использовать любой текстовый редактор, чтобы создать программу SCL, а затем импортировать этот файл в STEP 7.



В разделе интерфейса блока кода SCL могут быть объявлены следующие типы параметров:

- Input, Output, InOut и Ret_Val: Эти параметры определяют входные / выходные переменные (теги) и возвращаемое значение для блока кода. Имя переменной (тега), которое вводится здесь, используется локально во время исполнения блока кода. Обычно глобальное имя переменной (тега) не используется в таблице переменных (тегов).
- Static (только FB; на рисунке выше представлена FC): Блок кода используется статический переменные (теги), которые служат для сохранения статических промежуточных результатов в блоке данных экземпляра. Блок хранит статический данные до их перезаписи, т.е. частично в течение нескольких циклов. Вместе со статическими данными хранятся и имена блоков, которые будут вызываться в этом блоке кода как мультиэкземплярный DB.
- Temp: Эти параметры являются временными переменными (тегами), которые используются во время исполнения блока кода.
- Constant: Это именованные постоянные значения для блока кода.

При вызове блока кода SCL из другого блока кода, параметры блока кода SCL выступают как входы или выходы.



В этом примере переменные (теги) для "Start" и "On" (из таблицы переменных (тегов) системы) соответствуют переменным (тегам) "StartStopSwitch" и "RunYesNo" в таблице описания переменных программы SCL.

7.5.3.2 Выражения и операции SCL

Создание выражения SCL

Выражение SCL - это формула для вычисления значения. Выражение состоит из операндов и операторов (таких как *,/, + или-). Операнды могут быть переменными (тегами), константами или выражениями.

Оценка выражений происходит в определенной последовательности, задаваемой следующими факторами:

- У каждого оператора есть заранее определенный приоритет, при этом вначале выполняется операция с самым высоким приоритетом.
- В случае равного приоритета, операторы обрабатываются последовательно слева направо.
- С помощью круглых скобок обозначается последовательность операторов, которые будут вычислены вместе.

Результат выражения может использоваться для присвоения значения переменной (тегу), используемому в программе, как условие для управляющего оператора, как параметр для другой инструкции SCL или для вызова блока кода.

Таблица 7-2 Операторы в SCL

Тип	Операция	Оператор	Приоритет
Скобки	(*Выражение*)	(,)	1
Арифметика	Степень	**	2
	Знак (унарный плюс)	+	3
	Знак (унарный минус)	-	3
	Умножение	*	4
	Деление	/	4
	Модульная функция	MOD	4
	Сложение	+	5
	Вычитание	-	5
Сравнение	Меньше чем	<	6
	Меньше или равно	<=	6
	Больше чем	>	6
	Больше или равно	>=	6
	Равно	=	7
	Не равно	<>	7
Битовая логика	Отрицание (унарное)	NOT	3
	Логическая операция И	AND или &	8
	Логическая операция "Исключающее ИЛИ"	XOR	9
	Логическая операция ИЛИ	OR	10
Присваивание	Присваивание	:=	11

Как высокоуровневый язык программирования, SCL использует стандартные операторы для основных задач:

- Оператор присваивания: :=
- Арифметические функции: +, -, * и /
- Адресация глобальных переменных (тегов): "<имя переменной (тега)>" (имя переменной (тега) или имя блока данных, заключенное в двойные кавычки)
- Адресация локальных переменных: #<имя переменной> (имя переменной, которому предшествует символ "#")

Следующие примеры показывают различные выражения для различных случаев использования:

`"C" := #A+#B;`	Присваивает сумму двух локальных переменных тегу
`"Data_block_1".Tag := #A;`	Присваивание тегу блока данных
`IF #A > #B THEN "C" := #A;`	Условие для оператора IF-THEN
`"C" := SQRT (SQR (#A) + SQR (#B));`	Параметры для инструкции SQRT

Арифметические операторы могут обработать различные типы числовых данных. Тип данных результата будет определяться типом самого старшего операнда. Пример: Результатом операции умножения с операндом типа INT и операндом типа REAL является значение типа REAL.

Управляющие операторы

Управляющий оператор - это особый тип выражения SCL, которое выполняет следующие задачи:

- Ветвление программы
- Повторение разделов программного кода SCL
- Переход в другие части программы SCL
- Выполнение по условию

Управляющие операторы SCL это IF-THEN, CASE-OF, FOR-TO-DO, WHILE-DO, REPEAT-UNTIL, CONTINUE, GOTO иRETURN.

Одиночный оператор обычно занимает одну строку кода. Можно ввести несколько операторов в одну строку, или можно разбить оператор на несколько строк кода, чтобы улучшить читабельность кода. Разделители (такие как табуляция, разрывы строк и дополнительные пробелы) игнорируются при проверке синтаксиса. Оператор END завершает управляющий оператор.

Следующие примеры показывают управляющий оператор FOR-TO-DO. (Обе формы кодирования синтаксически допустимы.)

```
FOR x := 0 TO max DO sum := sum + value(x); END_FOR;
FOR x := 0 TO max DO
     sum := sum + value(x);
END_FOR;
```

Управляющий оператор может включаться в себя метку перехода. Метка перехода с двоеточием на конце устанавливается на начало оператора:

Метка перехода: <оператор>;

Система интерактивной помощи STEP 7 содержит всю справочную информацию по языку программирования SCL.

Условия

Условие - это выражение сравнения или логическое выражение, результат которого имеет тип BOOL (со значением ИСТИНА (TRUE) или ЛОЖЬ (FALSE)). Следующий пример показывает условия различных типов:

#Температура > 50 **#Счетчик <= 100** **#CHAR1 < 'S'**	Выражение отношения
(#Alpha <> 12) AND NOT #Beta	Выражение сравнения и логическое выражение
5 + #Alpha	Арифметическое выражение

Условие может использовать арифметические выражения:

- Условие выражения TRUE, если результатом является любое значение кроме нуля.
- Условие выражения FALSE, если результатом является ноль.

Вызов других блоков кода из программы SCL

Для вызова другого блока кода в программе пользователя, просто ввести имя (или абсолютный адрес) FB или FC с параметрами. Для FB необходимо указать DB экземпляра, который должен вызываться вместе с FB.

<имя DB> (список параметров)	Вызов как одиночного экземпляра
<имя экземпляра> (список параметров)	Вызов как мультиэкземпляра
"MyDB"(MyInput:=10, MyInOut:="Tag1");	
<имя FC> (список параметров)	Стандартный вызов
<операнд>:=<имя FC> (список параметров)	Вызов в выражении
"MyFC"(MyInput:=10, MyInOut:="Tag1");	

Также можно перетащить блоки мышью из дерева проекта в редактор текстов программы SCL программы и завершить параметрирование.

Добавление комментариев блока в код SCL

Можно добавить комментарий блока в код SCL, вставив его между (* и *). Между (* и *) может быть вставлено любое число строк комментария. Программный блок SCL может содержать несколько комментариев блока. Для упрощения программирования, редактор SCL содержит кнопку комментария блока, наряду с общими управляющими операторами:



Адресация

Также, как и LAD с FBD, SCL позволяет использовать либо переменные (теги) (символьная адресация), либо абсолютные адреса в программе пользователя. SCL позволяет использовать переменную и в качестве индекса массива.

Абсолютная адресация

`%I0.0` `%MB100`	Абсолютный адрес указывается после символа "%". Без "%" STEP 7 генерирует неопределенную ошибку переменной (тега) во время компиляции.

Символьная адресация

`"PLC_Tag_1"`	Переменная (тег) в таблице переменных (тегов) PLC
`"Data_block_1".Tag_1`	Переменная (тег) в блоке данных
`"Data_block_1".MyArray[#i]`	Элемент массива в массиве блока данных

7.5.3.3 Индексированная адресация с помощью инструкций PEEK и POKE

SCL предлагает инструкции PEEK и POKE, которые позволяют выполнять чтение/запись из/в блоки данных, ввод-вывод или память. Для операции указываются параметры для определенных байтовых или битовых смещений.

Примечание

Для использования инструкций PEEK и POKE с блоками данных потребуются стандартные (не оптимизированные) блоки данных. Также обратите внимание на то, что инструкции PEEK и POKE просто передают данные. У них нет информации о типах данных по указанным адресам.

`PEEK(area:=_in_,`
`dbNumber:=_in_,`
`byteOffset:=_in_);`
	Читает байт, на который ссылается byteOffset, из указанного блока данных, из I/O или области памяти.
	Пример ссылки на блок данных:
	`%MB100 := PEEK(area:=16#84, dbNumber:=1, byteOffset:=#i);`
	Пример ссылки на вход IB3:
	`%MB100 := PEEK(area:=16#81, dbNumber:=0, byteOffset:=#i); // when #i = 3`

`PEEK_WORD(area:=_in_,`
`dbNumber:=_in_,`
`byteOffset:=_in_);`
	Читает слово, на которое ссылается byteOffset, из указанного блока данных, из I/O или области памяти.
	Пример:
	`%MW200 := PEEK_WORD(area:=16#84, dbNumber:=1, byteOffset:=#i);`

```
PEEK_DWORD(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_);
```

Читает двойное слово, на которое ссылается byteOffset, из указанного блока данных, из I/O или области памяти.

Пример:

```
%MD300 := PEEK_DWORD(area:=16#84,
dbNumber:=1, byteOffset:=#i);
```

```
PEEK_BOOL(area:=_in_,
          dbNumber:=_in_,
          byteOffset:=_in_,
          bitOffset:=_in_);
```

Читает булево значение слово, на которое ссылается bitOffset и byteOffset, из указанного блока данных, из I/O или области памяти.

Пример:

```
%MB100.0 := PEEK_BOOL(area:=16#84,
dbNumber:=1, byteOffset:=#ii,
bitOffset:=#j);
```

```
POKE(area:=_in_,
     dbNumber:=_in_,
     byteOffset:=_in_,
     value:=_in_);
```

Записывает значение (байт, слово или двойное слово) в указанный byteOffset указанного блока данных, I/O или области памяти.

Пример ссылки на блок данных:

```
POKE(area:=16#84, dbNumber:=2,
byteOffset:=3, value:="Tag_1");
```

Пример ссылки на выход QB3:

```
POKE(area:=16#82, dbNumber:=0,
byteOffset:=3, value:="Tag_1");
```

```
POKE_BOOL(area:=_in_,
          dbNumber:=_in_,
          byteOffset:=_in_,
          bitOffset:=_in_,
          value:=_in_);
```

Записывает булево значение в указанный bitOffset и byteOffset указанного блока данных, I/O или области памяти.

Пример:

```
POKE_BOOL(area:=16#84, dbNumber:=2,
byteOffset:=3, bitOffset:=5,
value:=0);
```

```
POKE_BLK(area_src:=_in_,
         dbNumber_src:=_in_,
         byteOffset_src:=_in_,
         area_dest:=_in_,
         dbNumber_dest:=_in_,
         byteOffset_dest:=_in_,
         count:=_in_);
```

Записывает заданное в "count" число байтов, начиная с указанного byteOffset указанного исходного блока данных, исходного I/O или исходной области памяти в указанный byteOffset целевого блока данных, целевого I/O или целевой области памяти.

Пример:

```
POKE_BLK(area_src:=16#84,
dbNumber_src:=#src_db,
byteOffset_src:=#src_byte,
area_dest:=16#84,
dbNumber_dest:=#src_db,
byteOffset_dest:=#src_byte,
count:=10);
```

Для инструкций PEEK и POKE действуют следующие значения для параметров "area", "area_src" и "area_dest". Для областей, отличных от блоков данных, параметр dbNumber должен быть всегда равен 0.

16#81	I
16#82	Q
16#83	M
16#84	DB

7.5.4 EN и ENO в LAD, FBD и SCL

Определение "потока сигналов" (EN и ENO) для инструкции

Определенные инструкции (такие как математические инструкции и инструкции перемещения) обеспечивают параметры для EN и ENO. Эти параметры относятся к потоку сигналов в LAD или FBD и определяют, будет ли инструкция выполнена в этом цикле. В SCL также можно указать параметр ENO и для блока кода.

- EN (разрешающий вход) является булевым входом. На этом входе необходимо наличие потока сигналов (EN = 1), чтобы блок мог бы быть выполнен. Если вход EN блока LAD подключен напрямую к левой шине, то инструкция выполняется всегда.
- ENO (разрешающий выход) является булевым выходом. Если на входе EN блока имеет место поток сигналов и блок обрабатывается без ошибок, тогда выход ENO переводит поток сигналов (ENO = 1) на следующий элемент. Если при выполнении блока возникает ошибка, то поток энергии прерывается на блочной инструкции, ставшей причиной ошибки (ENO = 0).

Таблица 7-3 Операнды для EN и ENO

Редактор текстов программ	Входы/выходы	Операнды	Тип данных
LAD	EN, ENO	Поток сигналов	Bool
FBD	EN	E, E:P, A, M, DB, Temp, поток сигналов	Bool
	ENO	Поток сигналов	Bool
SCL	EN[1]	TRUE, FALSE	Bool
	ENO[2]	TRUE, FALSE	Bool

[1] Использование EN доступно только для FB.

[2] Использование ENO с блоком кода SCL является опциональным. Необходимо сконфигурировать компилятор SCL таким образом, чтобы ENO устанавливался при завершении обработки блока кода.

Конфигурирование установки ENO в SCL

Для конфигурирования компилятора SCL таким образом, чтобы ENO устанавливался, выполнить следующие шаги:

1. Выбрать в меню "Дополнительно" команду "Параметры".

2. Раскрыть свойства "Программирования PLC" и выбрать "SCL (Structured Control Language)".

3. Выбрать опцию "Автоматическая установка ENO".

Использование ENO в программном коде

Можно использовать ENO и в программном коде, например, присваивая ENO переменной (тегу) PLC, или путем обработки ENO в локальном блоке.

Примеры:

```
"MyFunction"
  ( IN1 := … ,
    IN2 := … ,
    OUT1 => #myOut,
    ENO => #statusFlag ); // PLC tag statusFlag holds the value of ENO

"MyFunction"
  ( IN1 := …
    IN2 := … ,
    OUT1 => #myOut,
    ENO => ENO );  // block status flag of "MyFunction"
                   // is stored in the local block

IF ENO = TRUE THEN
    // execute code only if MyFunction returns true ENO
```

Влияние параметров Ret_Val или Status на ENO

Некоторые инструкции, такие как коммуникационные инструкции или инструкции преобразования строк, имеют выходной параметр, который содержит информацию об обработке инструкции. У некоторых инструкций, напр., есть параметр Ret_Val (возвращаемое значение), обычно с типом данных Int, содержащий информацию о состоянии в диапазоне от -32768 до +32767. У других инструкций есть параметр Status, обычно с типом данных Word, хранящий информацию о состоянии в шестн. диапазоне от 16#0000 до 16#FFFF. Числовое значение в одном из параметров Ret_Val или Status указывает на состояние ENO соответствующей инструкции.

- Ret_Val: Значение между 0 и 32767 обычно устанавливает ENO = 1 (или TRUE). Значение между -32768 и -1 обычно устанавливает ENO = 0 (или FALSE). Для анализа Ret_Val следует изменить представление на шестнадцатеричное.
- Status: Значение между 16#0000 и 16#7FFF обычно устанавливает ENO = 1 (или TRUE). Значение между 16#8000 и 16#FFFF обычно устанавливает ENO = 0 (или FALSE).

Инструкции, которые выполняются в течение нескольких циклов, часто имеют параметр Busy (Bool), чтобы показать, что инструкция активна, но ее выполнение еще не завершено. У таких инструкций также часто есть параметр Done (Bool) и параметр Error

(Bool). Параметр Done показывает, что инструкций была выполнена без ошибок, а параметр Error указывает на то, что выполнение инструкции завершено с ошибками.

- Если Busy = 1 (или TRUE), то ENO = 1 (или TRUE).
- Если Done = 1 (или TRUE), то ENO = 1 (или TRUE).
- Если Error = 1 (или TRUE), то ENO = 0 (или FALSE).

См. также

Инструкции OK (проверить на достоверность) и NOT_OK (проверить на недостоверность) (Страница 244)

7.6 Защита

7.6.1 Защита от несанкционированного доступа для CPU

CPU предлагает четыре уровня защиты для ограничения доступа к определенным функциям. Установка уровней защиты и паролей для CPU разграничивает функции и области памяти, которые остаются доступными без ввода пароля.

Любой уровень защиты предоставляет неограниченный доступ к определенным функциям и без ввода пароля. Предустановка CPU - "без ограничений" и "без парольной защиты". Для защиты доступа к CPU, необходимо настроить свойства CPU и установить пароль.

Ввод пароля по сети не ставит под угрозу парольную защиту CPU. Защита паролем не распространяется на выполнение инструкций программы пользователя, включая коммуникационные функции. Ввод правильного пароля открывает доступ ко всем функциям, разрешенным для соответствующего уровня защиты.

Коммуникация между CPU (через функции связи в блоках кода) не ограничивается уровнем защиты CPU.

Таблица 7-4 Уровни защиты CPU

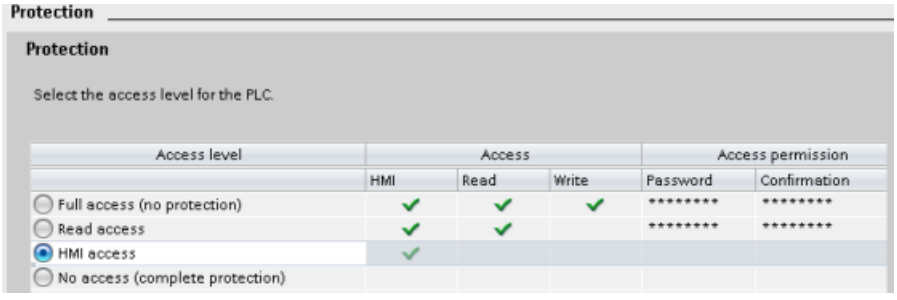
Уровень защиты	Ограничения доступа
Полный доступ (защита отсутствует)	Неограниченный доступ без парольной защиты.
Доступ по чтению	Разрешается доступ через HMI, сравнение блоков кода офлайн/онлайн и неограниченный обмен данными между CPU без парольной защиты. Для внесения изменений в CPU (обращения по записи) потребуется пароль. Для изменения рабочего состояния CPU (RUN/STOP) пароль не нужен.

Уровень защиты	Ограничения доступа
Доступ через HMI	Доступ через HMI и неограниченный обмен данными между CPU без парольной защиты. Для чтения данных в CPU, сравнение блоков кода офлайн/онлайн, изменения (записи) в CPU и для изменения рабочего состояния CPU (RUN/STOP) требуется пароль.
Доступ отсутствует (полная защита)	Нет доступа без парольной защиты. Для доступа через HMI, чтения данных в CPU, сравнение блоков кода офлайн/онлайн и для изменения (записи) данных в CPU требуется пароль.

Следует помнить, что на каждом уровне защиты можно установить (временный) аварийный IP-адрес (Страница 1016) для CPU.

Пароли чувствительны к регистру символов. Для конфигурирования уровня защиты и паролей, выполнить следующие действия:

1. Выбрать в "Конфигурации устройства" модуль CPU.
2. Открыть в окне инспектора вкладку "Свойства".
3. Выбрать свойство "Защита", чтобы установить уровень защиты и ввести пароли.



При загрузке этой конфигурации в CPU у пользователя есть доступ через HMI, и он может без пароля обращаться к функциям HMI. Для чтения данных или сравнения блоков кода офлайн/онлайн, пользователь должен ввести сконфигурированный для "Доступа по чтению" или для "Полного доступа (без защиты)" пароль. Для записи данных,

пользователь должен ввести сконфигурированный для "Полного доступа (без защиты)" пароль.

⚠ ПРЕДУПРЕЖДЕНИЕ

Несанкционированный доступ к защищенному CPU

У пользователей с полным доступом к CPU есть право читать и записывать переменные PLC. Независимо от уровня доступа для CPU у пользователей веб-сервера могут быть права на чтение и запись переменных PLC. Несанкционированный доступ к CPU или установка недопустимых значений для переменных PLC могут нарушить ход процесса и привести к тяжелым телесным повреждениям и/или материальному ущербу.

Авторизованные пользователи могут вносить изменения в рабочие состояния, записывать данные PLC и обновлять прошивку. Siemens рекомендует придерживаться следующих мер безопасности:

- Установить сложные пароли для уровней доступа CPU и ID пользователя веб-сервера (Страница 1028). Сложные пароли - это, по крайней мере, десять символов, представляющих комбинацию букв, цифр и специальных символов, не являющиеся словами, которые могут быть найдены в словаре, и не являющиеся именами или идентификаторами, которые могут быть получены из персональных данных. Следует хранить пароль в секрете и часто его изменять.
- Разрешить доступ к веб-серверу только по протоколу HTTPS.
- Не расширять минимальные права по умолчанию для пользователя веб-сервера "Все".
- Выполнить поиск ошибок и проверку диапазонов для переменных в программной логике, так как пользователи веб-страниц могут устанавливать для переменных PLC недопустимые значения.

Механизмы подключения

Для доступа к удаленным узлам с помощью инструкций PUT/GET пользователю также нужны соответствующие полномочия.

По умолчанию опция "Разрешить доступ с использованием коммуникации PUT/GET" отключена. В этом случае доступ по чтению и записи к данным CPU возможен только для коммуникационных соединений, которые требуют конфигурации или программирования как для локального CPU, так и для другого участника процесса коммуникации (партнера). Например, возможен доступ к помощью инструкций BSEND/BRCV.

Поэтому соединения, для которых локальный CPU является только сервером (подразумевается, что конфигурирование/программирование обмена с коммуникационным партнером на локальном CPU не выполняется), невозможны во время работы CPU. Это относится, напр., к:

- PUT/GET-, FETCH/WRITE- или FTP-доступу через коммуникационные модули
- PUT/GET-доступу из других S7-CPU
- доступу через HMI с использование PUT/GET-коммуникации

Если необходимо разрешить доступ к данным CPU со стороны клиента, т.е. не требуется ограничение служб коммуникации CPU, то действовать следующим образом:

1. Сконфигурировать для защиты от несанкционированного доступа любой уровень защиты, кроме "Доступ отсутствует (полная защита)".
2. Установить флажок "Разрешить доступ с использованием коммуникации PUT/GET".



После загрузки такой конфигурации в CPU, будет разрешена PUT/GET-связь с удаленными устройствами.

7.6.2 Внешняя загружаемая память

Можно заблокировать копирование из внутренней загружаемой памяти во внешнюю загружаемую память (карта памяти SIMATIC). Для блокировки копирования внутренней загружаемой памяти во внешнюю загружаемую память действовать следующим образом:

1. Выбрать в конфигурации устройства CPU в STEP 7 в разделе свойств "Общее" опцию "Защита".
2. Выбрать в разделе "Внешняя загружаемая память" опцию "Выключить копирование из внутренней загружаемой памяти во внешнюю загружаемую память".

В Вставка карты памяти в CPU (Страница 137) описано, как это свойство проявляется при вставке карты памяти в CPU.

7.6.3 Защита ноу-хау

С помощью защиты ноу-хау один или несколько блоков кода (OB, FB, FC или DB) можно защитить от несанкционированного доступа в программе. Для ограничения доступа к блоку кода следует использовать пароль. Пароль обеспечивает эффективную защиту от несанкционированного чтения или редактирования блока кода. Без пароля возможно лишь чтение следующей информации, относящейся к блоку кода:

- Название блока, комментарий и свойства блока
- Параметры передачи (IN, OUT, IN_OUT, возврат)
- Структура вызова программы
- Глобальные переменные (теги) в перекрестных ссылках (без информации об использования), но локальные переменные (теги) скрыты

Если блок конфигурируется для защиты "ноу-хау", то к коду внутри блока можно получить доступ только после ввода пароля.

Защита ноу-хау блока кода конфигурируется в окне задач "Свойства" соответствующего блока кода. После открытия блока кода выбрать "Защиту" в свойствах.



1. В свойствах блока кода нажать кнопку "Защита", чтобы открыть диалоговое окно "Защита ноу-хау".
2. Нажать кнопку "Определить", чтобы ввести пароль.



После ввода и подтверждения пароля, нажать "OK".



7.6.4 Защита от копирования

С помощью следующей функции обеспечения безопасности можно привязать программные блоки к определенной карте памяти или CPU. Эта функция предназначена в первую очередь для защиты интеллектуальной собственности. Привязка программного блока к определенному устройству позволяет использовать программу или блок кода только в комбинации с определенной картой памяти или CPU. Эта функция позволяет распространять программы / блоки кода в электронном виде (напр., через Интернет или по электронной почте) или посредством пересылки карты памяти. Защита от копирования доступна для OB (Страница 183), FB (Страница 185) и FC (Страница 184). S7-1200 CPU поддерживает три типа защиты блоков:

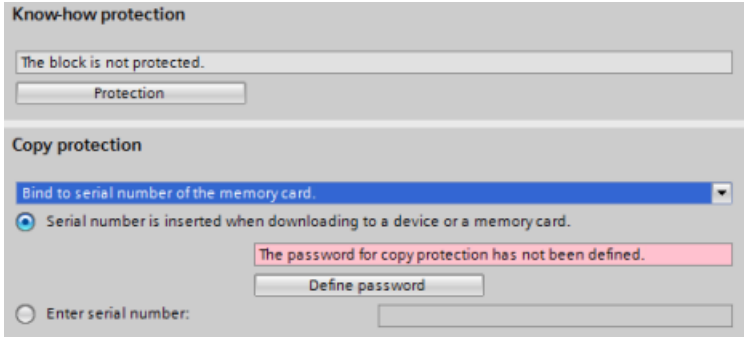
- Привязка к серийному номеру CPU
- Привязка к серийному номеру карты памяти
- Динамическая привязка с обязательным паролем

Для привязки блока к определенному CPU или карте памяти, открыть окно задач "Свойства" соответствующего блока кода.

1. После открытия блока кода выбрать "Защиту".



2. Выбрать в выпадающем списке "Защита от копирования" тип защиты от копирования, который будет использоваться.



3. Для привязки к серийному номеру CPU или карты памяти можно выбрать, будет ли серийный номер вставлен при загрузке или следует ввести серийный номер карты памяти или CPU.

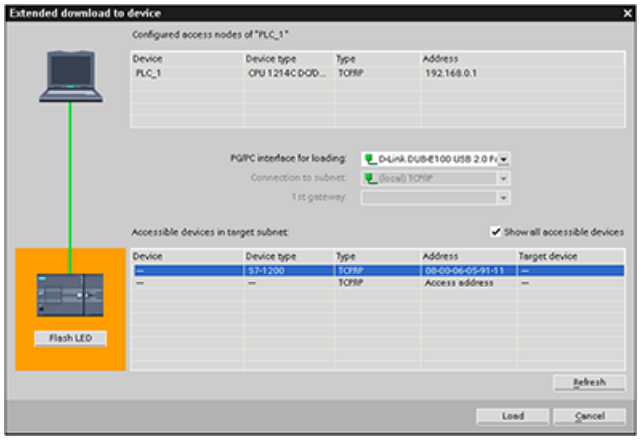
Примечание

Серийный номер чувствителен к регистру символов.

Для динамической привязки с обязательным паролем следует определить пароль, который будет использоваться для загрузки или копирования блока.
Впоследствии при загрузке (Страница 209) блока с динамической привязкой, потребуется ввести этот пароль, чтобы сделать загрузку возможной. Следует помнить, что пароль для защиты от копирования и пароль для защиты ноу-хау (Страница 206) - это два различных пароля.

7.7 Загрузка элементов программы

Можно загрузить элементы своего проекта из программатора в CPU. При загрузке проекта, CPU сохраняет программу пользователя (OB, FC, FB и DB) во внутренней загружаемой памяти или, при наличии карты памяти SIMATIC, во внешней загружаемой памяти (на карте).



Можно загрузить проект из программатора в CPU из одного из следующих мест:

- Дерево проекта: Кликнуть правой кнопкой мыши по элементу программы и выбрать "Загрузка в CPU".
- Онлайн-меню: Кликнуть по опции "Загрузка в устройство".
- Панель инструментов: Кликнуть по символу "Загрузка в устройство".
- Конфигурация устройства: Кликнуть правой кнопкой мыши по CPU и выбрать загружаемые элементы.

Если для программного блока установлена динамическая привязка с обязательным паролем (Страница 207), то для загрузки защищенных блоков потребуется ввести пароль. Если такой способ защиты от копирования был сконфигурирован для нескольких блоков, то потребуется ввод пароль для загрузки каждого из защищенных блоков.

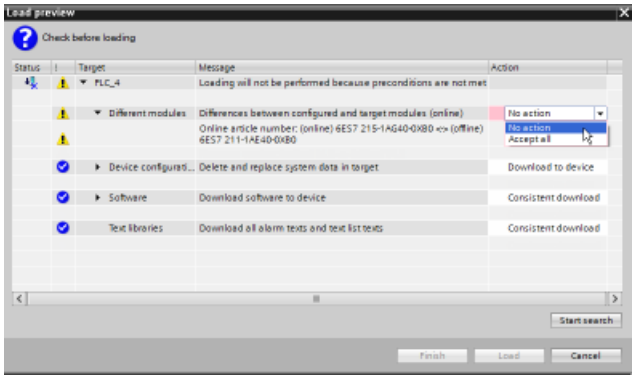
Примечание

Загрузка программы не удаляет и не изменяет значения в сохраняющей памяти. Для очистки сохраняющей памяти перед загрузкой следует сбросить CPU на заводские установки перед загрузкой программы.

Кроме этого, можно загрузить проект панели для базовых панелей HMI (Страница 32) из TIA Portal на карту памяти в S7-1200 CPU.

Загрузка в целевую систему, если сконфигурированный CPU отличается от подключенного CPU

STEP 7 и S7-1200 разрешают загрузку в целевую систему, если у подключенного CPU достаточно ресурсов для сохранения загружаемых из сконфигурированного CPU данных, исходя из требований памяти в проекте и совместимости периферии. Можно загрузить конфигурацию и программу из меньшего CPU в большее CPU, напр., из CPU 1211C DC/DC/DC в CPU 1215C DC/DC/DC, т.е. периферия является совместимой и памяти достаточно. В этом случае в процессе загрузки отображается предупреждение "Расхождение между сконфигурированными и целевыми модулями (онлайн)", а в диалоговом окне "Предпросмотр загрузки" - заказные номера и версии прошивки. Следует выбрать "Действия отсутствуют", если процесс загрузки должен быть прекращен, или "Принять все", если необходимо продолжить загрузку:



Примечание

Если после загрузки сконфигурированного CPU зайти онлайн (Страница 1390) в другой подключенный CPU, то проект для сконфигурированного CPU будет показан в дереве проекта с информацией о состоянии онлайн. Но в окне "Онлайн и диагностика" отображается фактический подключенный тип модуля CPU.

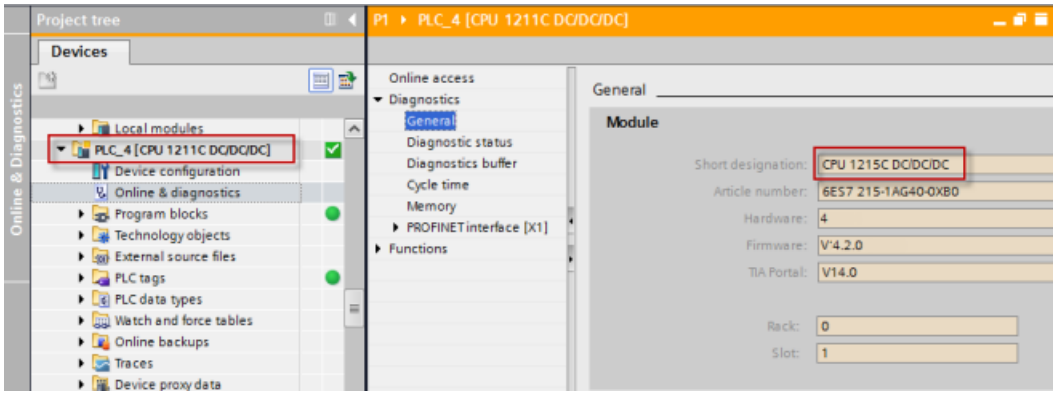


Рисунок 7-1 Онлайн просмотр, если сконфигурированный CPU отличается от подключенного CPU

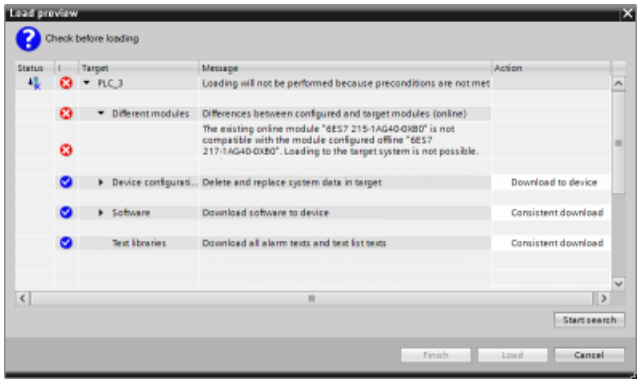
Конечно можно изменить свое устройство (Страница 166) в конфигурации устройства, чтобы тип модуля у сконфигурированного CPU совпал с типом модуля у подключенного

CPU. При попытке изменения устройства, в диалоговом окне "Изменение устройства" отображается вся информация о совместимости.

STEP 7 и S7-1200 блокируют загрузку в целевую систему, если у подключенного CPU недостаточно ресурсов для сохранения загружаемых из сконфигурированного CPU данных. Например, не получится загрузить конфигурацию оборудования и программу в следующих случаях:

- Из CPU 1215C DC/DC/DC в CPU 1212C DC/DC/DC, т.к. рабочей памяти недостаточно
- Из CPU 1211C DC/DC/реле в CPU 1211C DC/DC/DC, т.к. расхождения в периферии
- Из CPU 1217C DC/DC/DC в любой CPU 1211C, CPU 1212C, CPU 1214C или CPU 1215C, т.к. у CPU 1217C есть выходы 1,5 В
- Из CPU 1214C V4.2.x в CPU 1214C V4.0, т.е. версия прошивки не является совместимой "сверху вниз"

В таких случаях диалоговое окно "Предпросмотр загрузки" отображает ошибку:



Восстановление после неудачной загрузки в целевую систему

В случае неудачной загрузки, на вкладке "Информация" в окне инспектора отображается причина. Буфер диагностики также содержит информацию. После неудачной попытки загрузки, действовать следующим образом, чтобы выполнить успешную загрузку в целевую систему:

1. Устранить проблему согласно описанию в сообщении об ошибке.
2. Повторить попытку загрузки данных в целевую систему.

В редких случаях процесс загрузки завершается успешно, но последующее выключение и включение CPU не удается. В этом случае в буфере диагностики может находиться схожая со следующей ошибка:

- 16# 02:4175 -- Ошибка CPU: Ошибка анализа карты памяти: Неизвестная или несовместимая версия актуального типа карты в описании конфигурации CPU: Нет завершенной/отмененной функции карты памяти, установлена новая блокировка пуска: ..- Нет карты памяти, неправильный тип, неправильное содержание или защищенная карта

Если возникает такая ситуация и последующие попытки загрузки оканчиваются неудачей, следует очистить внутреннюю или внешнюю загружаемую память:

1. При работе с внутренней загружаемой памятью, сбросить CPU на заводские установки.
2. При работе с картой памяти SIMATIC, извлечь ее и стереть содержание карты памяти (Страница 142). После снова вставить карту.
3. Загрузить конфигурацию оборудования и ПО в целевую систему.

См. также

Синхронизация Онлайн-CPU и Офлайн-проекта (Страница 212)

7.8 Синхронизация Онлайн-CPU и Офлайн-проекта

При загрузке блоков проекта в CPU, он может определить, произошло ли изменение блоков или переменных (тегов) в Онлайн-CPU с момента последней загрузки. После CPU предложит синхронизировать изменения. Это означает, что можно загрузить изменения из Онлайн-CPU в проект, прежде чем проект будет загружен в CPU. Изменения в Онлайн-CPU могут быть обусловлены множеством факторов:

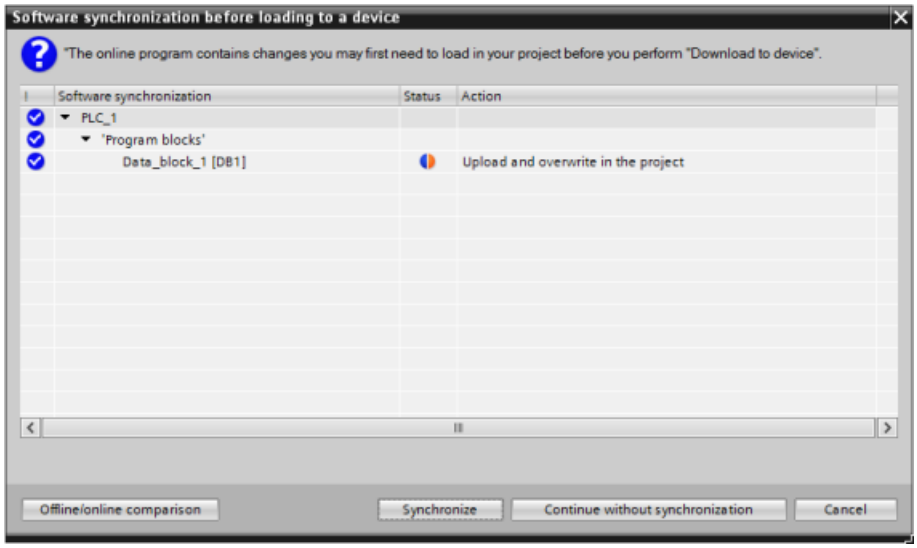
- Изменения стартовых значений переменных (тегов) блоков данных при работе, напр., через инструкцию WRIT_DBL (Страница 541) или через загрузку рецепта
- Процесс загрузки из "вторичного" проекта (проект, отличный от проекта последней загрузки), при котором присутствует одно или несколько из следующих условий:
  - Онлайн-CPU содержит программные блоки, отсутствующие в проекте.
  - Переменные (теги) блоков данных или атрибуты блоков отличаются в Офлайн-проекте и в Онлайн-CPU.
  - В Онлайн-CPU есть переменные (теги) PLC, отсутствующие в Офлайн-проекте.

Примечание

Если в проекте, использовавшемся для последнего процесса загрузки, обрабатываются блоки или переменные (теги), то выбор в контексте синхронизации не требуется. STEP 7 и CPU определяют, что изменения в Офлайн-проекте более новые, чем Онлайн-CPU, и продолжают обычный процесс загрузки.

Опции синхронизации

При загрузке проекта в CPU отображается диалоговое окно синхронизации, если STEP 7 определяет, что блоки данных или переменные (теги) в Онлайн-CPU новее, чем значения в проекте. Пример: Если программа STEP 7 выполнила WRIT_DBL и изменила стартовое значение для переменной (тега) в Data_block_1, то сразу после запуска загрузки в целевую систему STEP 7 отображает следующее диалоговое окно синхронизации:



В этом диалоговом окне перечисляются программные блоки, в которых имеются расхождения. В этом диалоговом окне предлагаются следующие возможности:

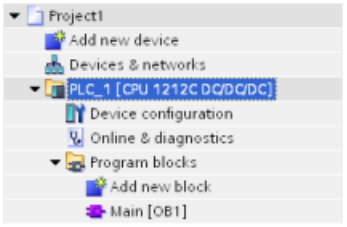
- Сравнение онлайн/оффлайн: Если нажать эту кнопку, то STEP 7 отобразит программные блоки, системные блоки, технологические объекты, переменные (теги) PLC и типы данных PLC для проекта как длясравнения с Онлайн-CPU (Страница 1399). Кликнув по объекту, можно отобразить подробный анализ отличий, включая отметки времени. С помощью этой информации можно решить, как поступить с расхождениями между Онлайн-CPU и проектом.
- Синхронизация: Если нажать на эту кнопку, то STEP 7 выгрузит блоки данных, переменные (теги) и другие объекты из Онлайн-CPU в проект. После можно загрузить программу в целевую систему, только если выполнение программы снова не привело к рассинхронизации проекта с CPU.
- Продолжить без синхронизации: Если нажать на эту кнопку, то STEP 7 загрузит проект в CPU.
- Отмена: Если нажать на эту кнопку, то произойдет отмена загрузки.

7.9 Выгрузка из Онлайн-CPU

Можно копировать программные блоки из Онлайн-CPU или подключенной к программатору карты памяти.

Подготовить офлайн проект для копируемых программных блоков:

1. Добавить модуль CPU, соответствующий Онлайн-CPU.
2. Раскрыть узел CPU так, чтобы была видна папка "Программные блоки".



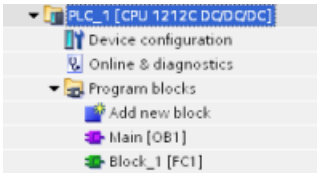
Для выгрузки программных блоков из Онлайн-CPU в Офлайн-проект, выполнить следующие действия:





1. Кликнуть в Офлайн-проекте по папке "Программные блоки".
2. Нажать кнопку "Установить онлайн-соединение".
3. Нажать кнопку "Выгрузка из CPU"
4. Подтвердить процесс в диалоговом окне Выгрузка из CPU (Страница 1390).

Когда выгрузка завершена, STEP 7 отображает все выгруженные программные блоки в проекте



7.9.1 Сравнение Онлайн-CPU с Офлайн-CPU

С помощью редактора "Сравнение" (Страница 1399) в STEP 7 можно найти различия между Онлайн- и Офлайн-проектом. Это может быть полезным до выгрузки программы из CPU.

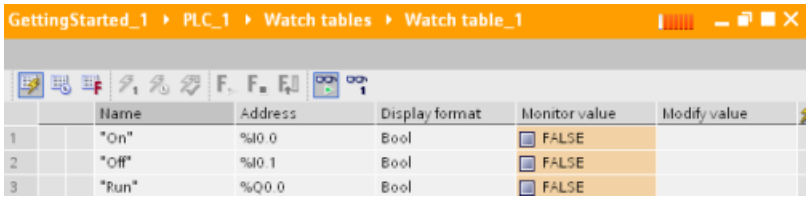
7.10 Отладка и тестирование программы

7.10.1 Мониторинг и изменение данных в CPU

Можно контролировать и изменять значения в Онлайн-CPU (см. следующую таблицу).

Таблица 7-5 Мониторинг и изменение данных с помощью STEP 7

Редактор	Мониторинг	Изменение	Принудительное присваивание значений
Таблица текущего контроля	Да	Да	Нет
Таблица принудительно задаваемых значений	Да	Нет	Да
Редактор текстов программ	Да	Да	Нет
Таблица переменных (тегов)	Да	Нет	Нет
Редактор блоков данных	Да	Нет	Нет



Мониторинг в таблице текущего контроля



Мониторинг в LAD-редакторе

В главе "Онлайн и диагностика" можно найти дополнительную информацию по теме Мониторинг и изменение данных в CPU (Страница 1401).

7.10.2 Таблицы текущего контроля и таблицы принудительно задаваемых значений

Для мониторинга и изменения значений программы пользователя, выполняемой Онлайн-CPU, можно использовать "таблицы текущего контроля". Можно создавать и сохранять в своем проекте различные таблицы текущего контроля для поддержки разнообразных сред тестирования. Это дает вам возможность выполнять тестирование при вводе в эксплуатацию или для целей обслуживания и текущего ремонта.

С помощью таблицы текущего контроля можно контролировать, как CPU выполняет программу пользователя, и вмешиваться в этот процесс. Можно вызывать и изменять значения не только для переменных (тегов) блоков кода и блоков данных, но также и для областей памяти CPU, включая входы и выходы (I и Q), периферийные входы (I:P), битовую память (M) и блоки данных (DB).

С помощью таблицы текущего контроля можно разблокировать физические выходы (Q:P) модуля CPU, находящегося в состоянии STOP. Например, можно назначать конкретные значения выходам при тестировании электрических соединений CPU.

Кроме этого STEP 7 предлагает таблицу принудительно задаваемых значений для принудительного присваивания определенного значения переменной (тегу). В главе "Онлайн и диагностика" в разделе Принудительное присваивание значений в CPU (Страница 1409) можно найти дополнительную информацию о принудительном присваивании значений.

Примечание

Принудительно задаваемые значения сохраняются в CPU, а не в таблице текущего контроля.

Нельзя принудительно присвоить значение входу (адрес "E"). Но можно принудительно присвоить значение периферийному входу. Для принудительного присваивания значения периферийному входу, прикрепить ":P" к адресу (пример: "On:P").

Кроме этого, STEP 7 позволяет отслеживать и записывать переменные программы на основании условий запуска (Страница 1421).

7.10.3 Перекрестная ссылка для отображения использования

Окно инспектора отображает информацию по перекрестным ссылкам, т.е. как выбранный объект используется во всем проекте, напр., в программе пользователя, в CPU или устройствах HMI. Вкладка "Перекрестная ссылка" отображает экземпляры, где используется выбранный объект и другие объекты используют его Окно инспектора также содержит блоки, которые доступны только онлайн в перекрестных ссылках. Чтобы отобразить перекрестные ссылки, выбрать команду "Показать перекрестные ссылки". (В проектном представлении эта команда находится в меню "Инструменты".)

Примечание

Не нужно закрывать редактор, чтобы увидеть информацию по перекрестным ссылкам.

Записи в перекрестных ссылках можно сортировать. Список перекрестных ссылок предоставляет обзор используемых в программе пользователя адресов ячеек памяти и переменных (тегов).

- При создании или редактировании программы сохраняется обзор используемых операндов, переменных (тегов) и вызовов блоков.
- Из перекрестных ссылок можно перейти непосредственно к месту использования операндов и переменных (тегов).
- Во время теста программы или при устранении ошибок поступают уведомления о том, какая ячейка памяти обрабатывается, какой командой, в каком блоке, какая используется переменная (тег), на каком экране и какой из блоков вызывается каким-либо другим блоком.

Таблица 7-6 Элементы перекрестных ссылок

Столбец	Описание
Объект (Object)	Имя объекта, который использует указанные объекты нижнего уровня или используется таковыми.
Количество (Number)	Количество использований
Использование (Point of use)	Месть использования, напр., сегмент
Свойство (Property)	Специальные свойства адресуемых объектов, например, имена переменных (тегов) в описаниях мультиэкземпляров
В качестве (as)	Показывает дополнительную информацию об объекте, например, используется ли DB экземпляра в качестве шаблона или как мультиэкземпляр
Доступ (Access)	Тип доступа, т.е. является ли доступ к операнду доступом по чтению (R) и/или доступом по записи (W)
Адрес (Address)	Адрес операнда
Тип (Type)	Информация о типе и языке, используемых при создании объекта
Путь (Path)	Месторасположение объекта в дереве проекта

В зависимости от установленных продуктов, таблица перекрестных ссылок выводит на экран дополнительные или отличные столбцы.

7.10.4 Структура вызовов для проверки иерархии вызовов

Структура вызовов показывает иерархиею вызовов блока в программе пользователя. Она предлагает обзор использованных блоков, вызовы других блоков, отношения между блоками, требования к данным для каждого блока и состояние отдельных блоков. Блоки в структуре вызовов могут открываться и редактироваться с помощью редактора текстов программы.

При отображении структуры вызовов пользователь видит список используемых в программе пользователя блоков. STEP 7 выделяет первый уровень структуры вызовов и отображает в том числе и блоки, которые не вызываются никаким другим блоком в программе. Первый уровень структуры вызовов содержит OB и те FC, FB и DB, которые не вызываются через OB. Если блок кода вызывает другой блок, то вызываемый блок

отображается с отступом под вызывающим блоком. Структура вызовов отображает только блоки, вызываемые блоком кода.

Можно выборочно показать только те блоки, которые являются причиной конфликтов в структуре вызовов. Следующие условия вызывают конфликты:

- Блоки, которые выполняют любые вызовы с более старыми или более новыми отметками времени в коде
- Блоки, которые вызывают блок с измененным интерфейсом
- Блоки, которые используют переменную (тег) с измененным адресом и/или измененным типом данных
- Блоки, которые ни прямо, ни косвенно не вызываются OB.
- Блоки, которые вызывают несуществующий или отсутствующий блок

Можно объединить несколько вызовов блоков и блоков данных в группу. Через выпадающий список можно отобразить ссылки на различные места вызовов.

Также можно выполнить проверку непротиворечивости, чтобы выявить конфликты по отметкам времени. Изменение отметки времени блока во время или после генерации программы, может привести к конфликтам по отметкам времени. Такие конфликты приводят к несогласованности между вызывающими и вызываемыми блоками.

- Большая часть конфликтов по отметкам времени и по интерфейсам может быть исправлена путем повторной компиляции блоков кода.
- Если компиляции не устраняет несогласованность, следует использовать информацию в столбце "Подробности" для перехода к источнику проблемы в редакторе текста программы. Здесь можно устранить противоречия вручную.
- Должны быть перекомпилированы все блоки, отмеченные красным цветом.

Инструкции

8

8.1 Битовые логические операции

8.1.1 Инструкции битовой логики

Языки LAD и FBD являются эффективными инструментами для работы с булевой логикой. Хотя преимущества языка SCL в первую очередь проявляются при работе со сложными математическими вычислениями и проектными структурами, он может использоваться и для работы с булевой логикой.

LAD контакты

Таблица 8-1 Нормально разомкнутые (NO) и нормально замкнутые контакты (NC)

LAD	SCL	Описание

"IN" 	`IF in THEN`
	`  Statement;`
	`ELSE`
	`  Statement;`
	`END_IF;`
		Нормально разомкнутые (NO) и нормально замкнутые контакты (NC): Можно соединять контакты друг с другом, создавая собственную комбинационную логику. Если указанный входной бит использует идентификатор памяти I (вход) или Q (выход), то значение этого бита считывается из регистра образа процесса. Физические сигналы с контактов в управляющем процессе подключаются к клеммам I на PLC. CPU опрашивает подключенные входные сигналы и непрерывно обновляет соответствующие значения состояний в образе процесса на входах.
		Прямое считывание физического входа выполняется с помощью символа ":P" после I адреса (пример: "%I3.4:P"). При прямом считывании значения битовых данных снимаются непосредственно с физического входа, а не из образа процесса. При прямом считывании образ процесса не обновляется.

"IN" 	`IF NOT (in) THEN`
	`  Statement;`
	`ELSE`
	`  Statement;`
	`END_IF;`	

Таблица 8-2 Тип данных для параметров

Параметры	Тип данных	Описание
IN	Bool	Назначенный бит

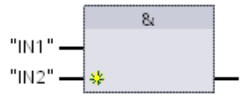
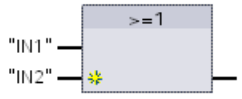
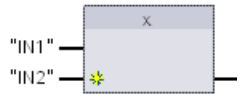
- Замыкающий (NO) контакт замкнут (ON), если назначенное значение бита равно 1.
- Размыкающий (NC) контакт замкнут (ON), если назначенное значение бита равно 0.
- Контакты, соединенные последовательно, образуют логическое И.
- Контакты, соединенные параллельно, образуют логическое ИЛИ.

Блоки AND (И), OR (ИЛИ) и XOR (исключающее ИЛИ) в FBD

При программировании на языке FBD, сегменты с LAD-контактами преобразуются в сегменты с блоками И (AND, &), ИЛИ (OR, >=1) и исключающее ИЛИ (XOR, x), в которых можно задавать значения битов для входов и выходов блока. Также можно создавать соединения с другими логическими блоками, образуя свою собственную комбинационную логику. После размещения блока в сегменте, можно перетащить функцию "Вставить вход" из панели инструментов "Избранное" или из дерева инструкцию на сторону входов блока для добавления входов. Также можно кликнуть правой кнопкой мыши по входному соединителю блока и выбрать "Вставить вход".

Входы и выходы блока могут быть соединены с другими блоками, или можно ввести адрес бита или символьное имя бита для неподключенного входа. При выполнении блочной инструкции, текущие состояния входов обрабатываются согласно двоичной логике блока, и если значение истинно, то выход будет также истинным.

Таблица 8-3 Блоки И, ИЛИ и ИСКЛЮЧАЮЩЕЕ ИЛИ

FBD	SCL [1]	Описание
	`out := in1 AND in2;`	Для того, чтобы выход был TRUE, все входы блока AND должны быть TRUE
	`out := in1 OR in2;`	Для того, чтобы выход был TRUE, любой вход блока OR должны быть TRUE
	`out := in1 XOR in2;`	Для того, чтобы выход был TRUE, нечетное число входов блока XOR должны быть TRUE

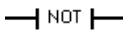
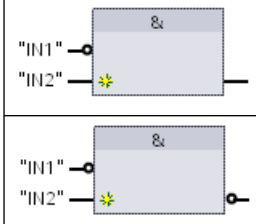
[1] В SCL: Необходимо присвоить результат операции переменной, чтобы можно было бы использовать его в другом операторе.

Таблица 8-4 Тип данных для параметров

Параметр	Тип данных	Описание
IN1, IN2	Bool	Входной бит

Логический элемент НЕ (NOT)

Таблица 8-5 Инверсия RLO (результата логической операции)

LAD	FBD	SCL	Описание
		**NOT**	При работе с языком программирования FBD, можно перетащить функцию "Инвертировать RLO" из панели инструментов "Избранное" или из дерева инструкций на вход или выход для создания логического элемента НЕ на данном соединителе блока. LAD-контакт NOT инвертирует логическое состояние потока сигналов на входе. • Если нет потока сигналов на NOT-контакт, значит имеет место поток сигналов на выходе. • Если есть поток сигналов на NOT-контакт, значит нет потока сигналов на выходе.

Выходная катушка и блок назначения

Инструкция для выхода катушки записывает значения в выходной бит. Если указанный выходной бит использует идентификатор памяти Q, то CPU включает или выключает выходной бит в регистре образа процесса, устанавливая указанный бит в соответствии с потоком сигналов. Выходные сигналы для исполнительных устройств подключаются к выходным клеммам CPU. В рабочем состоянии RUN модуль CPU непрерывно опрашивает входные сигналы, обрабатывает входные состояния в соответствии с логикой программы, а затем реагирует на них, устанавливая новые выходные значения в образе процесса выходов. CPU передает хранящуюся в регистре образа процесса реакцию на новое выходное состояние на подключенные выходные клеммы.

Таблица 8-6 Назначение и отрицание назначения

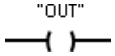
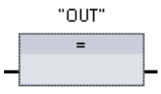
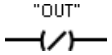
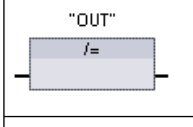
LAD	FBD	SCL	Описание
		**out := <булево выражение>;**	При работе с языком программирования FBD, LAD катушки преобразуются в блоки назначения (= и /=), при этом должен быть задан адрес бита для выхода блока. Входы и выходы блока могут быть соединены с другими логическими блоками, или можно ввести адрес бита. Прямая запись на физический выход выполняется с помощью символа ":P" после Q адреса (пример: "%Q3.4:P"). При прямой записи битовые значения записываются на выход в образе процесса и непосредственно на физический выход.
		**out := NOT <булево выражение>;**	

Таблица 8-7 Типы данных для параметров

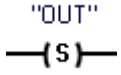
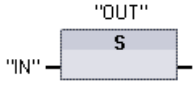
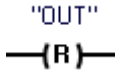
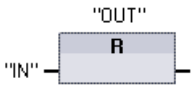
Параметр	Тип данных	Описание
OUT	Bool	Назначенный бит

- Если имеет место поток сигналов через выходную катушку или FBD-блок "=" активирован, то выходной бит устанавливается на 1.
- Если нет потока сигналов через выходную катушку или нет активированного FBD-блока "=", то выходной бит устанавливается на 0.
- Если имеет место поток сигналов через инвертированную выходную катушку или FBD-блок "/=" активирован, то выходной бит устанавливается на 0.
- Если нет потока сигналов через инвертированную выходную катушку или нет активированного FBD-блока "/=", то выходной бит устанавливается на 1.

8.1.2 Операции установки и сброса

Установка и сброс 1 бита

Таблица 8-8 Инструкции S и R

LAD	FBD	SCL	Описание
		Недоступно	Установка выхода: При активации S (установить), значение данных на адресе OUT устанавливается на 1. Если S активируется, то OUT не изменяется.
		Недоступно	Сброс выхода: При активации R (сбросить), значение данных на адресе OUT устанавливается на 0. Если R активируется, то OUT не изменяется.

[1] В LAD и FBD: Данные инструкции могут быть вставлены в любом месте сегмента.

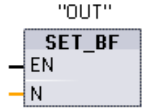
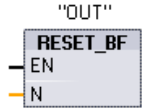
[2] В SCL: Необходимо написать код для репликации данной функции в приложении.

Таблица 8-9 Типы данных для параметров

Параметр	Тип данных	Описание
IN (или соединение с контактной или блочной логикой)	Bool	Битовая переменная (тег) контролируемого адреса
OUT	Bool	Битовая переменная (тег) устанавливаемого или сбрасываемого адреса

Установка и сброс битового поля

Таблица 8-10 Инструкции SET_BF и RESET_BF

LAD[1]	FBD	SCL	Описание
		Недоступно	Установка битового поля: При активации SET_BF значение данных 1 записывается в "n" битов, начиная с адреса OUT. Если SET_BF не активируется, то OUT не изменяется.
		Недоступно	Сброс битового поля: RESET_BF записывает значение данных 0 записывается в "n" битов, начиная с адреса OUT. Если RESET_BF не активируется, то OUT не изменяется.

[1] В LAD и FBD: Эти инструкции должны располагаться на самом правом краю ветви.

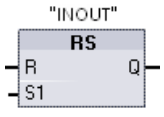
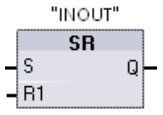
[2] В SCL: Необходимо написать код для репликации данной функции в приложении.

Таблица 8-11 Типы данных для параметров

Параметр	Тип данных	Описание
OUT	Bool	Начальный элемент битового поля для установки или сброса (пример: #MyArray[3])
n	Константа (UInt)	Количество записываемых битов

Триггеры с приоритетом установки и сброса

Таблица 8-12 Инструкции RS и SR

LAD/FBD	SCL	Описание
	Недоступно	Триггер сброса/установки: RS это триггер с приоритетом установки. Если оба сигнала установки (S1) и сброса (R) истинны, то выходной адрес INOUT устанавливается на 1.
	Недоступно	Триггер сброса/установки: SR это триггер с приоритетом сброса. Если оба сигнала установки (S) и сброса (R1) истинны, то выходной адрес INOUT устанавливается на 0.

[1] В LAD и FBD: Эти инструкции должны располагаться на самом правом краю ветви.

[2] В SCL: Необходимо написать код для репликации данной функции в приложении.

Таблица 8-13 Типы данных для параметров

Параметр	Тип данных	Описание
S, S1	Bool	Установка входа; 1 означает приоритет
R, R1	Bool	Сброс входа; 1 означает приоритет

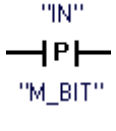
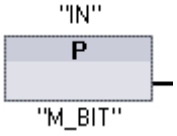
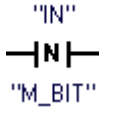
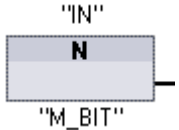
Параметр	Тип данных	Описание
INOUT	Bool	Назначенная битовая переменная (тег) "INOUT"
Q	Bool	Следит за состоянием бита "INOUT"

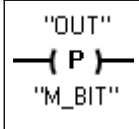
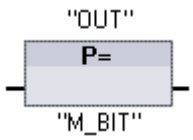
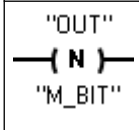
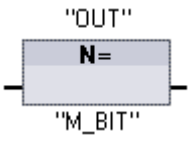
Переменная (тег) "INOUT" указывает адрес бита для установки или сброса.
Опциональный выход Q указывает состояние сигнала для адреса "INOUT".

Инструкция	S1	R	Бит "INOUT"
RS	0	0	Предыдущее состояние
	0	1	0
	1	0	1
	1	1	1
	S	**R1**	
SR	0	0	Предыдущее состояние
	0	1	0
	1	0	1
	1	1	0

8.1.3 Операции по положительному и отрицательному фронту

Таблица 8-14 Инструкции "Положительный фронт" и "Отрицательный фронт"

LAD	FBD	SCL	Описание
		Недоступно [1]	Опрос операнда на положительный фронт сигнала. LAD: Состояние этого контакта TRUE, если положительный фронт (переход из ВЫКЛ во ВКЛ) обнаруживается на назначенном бите IN. Логическое состояние этого контакта затем сопрягается с состоянием потока сигнала на входе, чтобы установить состояние потока сигнала на выходе. Контакт P может быть расположен в любом месте сегмента, за исключением конца ветви. FBD: Логическое состояние выхода TRUE, если положительный фронт (переход из ВЫКЛ во ВКЛ) обнаруживается на назначенном входном бите. Блок P может располагаться только в начале ветви.
		Недоступно [1]	Опрос операнда на отрицательный фронт сигнала. LAD: Состояние этого контакта TRUE, если отрицательный фронт (переход из ВКЛ во ВЫКЛ) обнаруживается на назначенном входном бите. Логическое состояние этого контакта затем сопрягается с состоянием потока сигнала на входе, чтобы установить состояние потока сигнала на выходе. Контакт N может быть расположен в любом месте сегмента, за исключением конца ветви. FBD: Логическое состояние выхода TRUE, если отрицательный фронт (переход из ВКЛ во ВЫКЛ) обнаруживается на назначенном входном бите. Блок N может располагаться только в начале ветви.

LAD	FBD	SCL	Описание
		Недоступно [1]	Установка операнда при положительном фронте сигнала. LAD: Назначенный бит OUT имеет состояние TRUE, если положительный фронт (переход из ВЫКЛ во ВКЛ) обнаруживается на потоке сигнала на входе катушки. Состояние потока сигнала на входе всегда проходит через катушку как состояние потока сигнала на выходе. Катушка P может быть вставлена в любом месте сегмента. FBD: Назначенный бит OUT имеет состояние TRUE, если положительный фронт (переход из ВЫКЛ во ВКЛ) обнаруживается на логическом состоянии входного соединителя блока или на назначенном входном бите, если блок находится в начале ветви. Логическое состояние на входе всегда проходит через блок как логическое состояние на выходе. Блок P= может быть вставлен в любом месте ветви.
		Недоступно [1]	Установка операнда при отрицательном фронте сигнала. LAD: Назначенный бит OUT имеет состояние TRUE, если отрицательный фронт (переход из ВЫКЛ во ВКЛ) обнаруживается на потоке сигнала на входе катушки. Состояние потока сигнала на входе всегда проходит через катушку как состояние потока сигнала на выходе. Катушка N может быть вставлена в любом месте сегмента. FBD: Назначенный бит OUT имеет состояние TRUE, если отрицательный фронт (переход из ВКЛ во ВЫКЛ) обнаруживается на логическом состоянии входного соединителя блока или на назначенном входном бите, если блок находится в начале ветви. Логическое состояние на входе всегда проходит через блок как логическое состояние на выходе. Блок N= может быть вставлен в любом месте ветви.

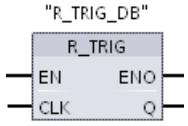
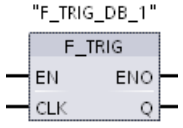
[1] В SCL: Необходимо написать код для репликации данной функции в приложении.

Таблица 8-15 P_TRIG и N_TRIG

LAD/FBD	SCL	Описание
	Недоступно [1]	Опрос RLO (результат логической операции) на положительный фронт сигнала. Поток сигналов или логическое состояние выхода Q принимает значение TRUE при обнаружении положительного фронта (переход из ВЫКЛ во ВКЛ) на состоянии входа CLK (FBD) или на потоке сигналов CLK (LAD). В LAD инструкция P_TRIG не может находиться в начале или конце сегмента. В FBD инструкция P_TRIG может быть расположена в любом месте, за исключением конца ветви.
	Недоступно [1]	Опрос RLO на отрицательный фронт сигнала. Поток сигналов или логическое состояние выхода Q принимает значение TRUE при обнаружении отрицательного фронта (переход из ВКЛ во ВЫКЛ) на состоянии входа CLK (FBD) или на потоке сигналов CLK (LAD). В LAD инструкция N_TRIG не может находиться в начале или конце сегмента. В FBD инструкция N_TRIG может быть расположена в любом месте, за исключением конца ветви.

[1] В SCL: Необходимо написать код для репликации данной функции в приложении.

Таблица 8-16 Инструкции R_TRIG и F_TRIG

LAD/FBD	SCL	Описание
	**"R_TRIG_DB"(CLK:=_in_, Q=> _bool_out_);**	Установка переменной (тега) по положительному фронту сигнала. Предшествующее состояние входа CLK сохраняется в назначенном DB экземпляра. Поток сигналов или логическое состояние выхода Q принимает значение TRUE при обнаружении положительного фронта (переход из ВЫКЛ во ВКЛ) на состоянии входа CLK (FBD) или на потоке сигналов CLK (LAD). В LAD инструкция R_TRIG не может находиться в начале или конце сегмента. В FBD инструкция R_TRIG может быть расположена в любом месте, за исключением конца ветви.
	"F_TRIG_DB"(CLK:=_in_, Q=> _bool_out_);	Установка переменной (тега) по отрицательному фронту сигнала. Предшествующее состояние входа CLK сохраняется в назначенном DB экземпляра. Поток сигналов или логическое состояние выхода Q принимает значение TRUE при обнаружении отрицательного фронта (переход из ВКЛ во ВЫКЛ) на состоянии входа CLK (FBD) или на потоке сигналов CLK (LAD). В LAD инструкция F_TRIG не может находиться в начале или конце сегмента. В FBD инструкция F_TRIG может быть расположена в любом месте, за исключением конца ветви.

Для R_TRIG иF_TRIG при вставке инструкции в программу автоматически открывается диалоговое окно "Опции вызова". В этом диалоговом окне можно определить, будет ли меркер фронта сохранен в собственном блоке данных (отдельный экземпляр) или в виде локальной переменной/тега (мультиэкземпляр) на интерфейсе блока. После создания отдельный блок данных можно найти в дереве проекта в папке "Ресурсы программы" по адресу "Программные модули > Системные модули".

Таблица 8-17 Типы данных для параметров (контакты/катушкиP иN, P=, N=, P_TRIG and N_TRIG)

Параметр	Тип данных	Описание
M_BIT	Bool	Меркер, в который сохраняется предшествующее состояние входа
IN	Bool	Входной бит, фронт которого обнаруживается
OUT	Bool	Выходной бит, указывающий, что был обнаружен фронт
CLK	Bool	Поток сигналов или входной бит, фронт которого обнаруживается
Q	Bool	Выход, указывающий, что был обнаружен фронт

Все инструкции выделения фронта используют меркер (M_BIT: P/N контакты/катушки, P_TRIG/N_TRIG) или (бит DB экземпляра: R_TRIG, F_TRIG) для сохранения предшествующего состояния контролируемого входного сигнала. Фронт обнаруживается путем сравнения состояния входа с предшествующим состоянием. Если состояния на входе указывают на изменение сигнала в нужном направлении, то

сигнализируется фронт, при этом выход устанавливается на TRUE. В иных случаях выход устанавливается на FALSE.

Примечание

Операции по фронту анализируют вход и значения меркера при каждом выполнении, в том числе и при первом выполнении. В программе должны быть учтены исходные состояния входа и меркера, чтобы разрешить обнаружение фронта в первом цикле или нет.

Так как меркер должен сохраняться от одного выполнения до следующего, для каждой операции по фронту следует использовать однозначный бит. Нельзя использовать этот бит в других местах в программе. Также избегать временной памяти и памяти, которую могут изменять другие системные функции, напр., обновления I/O. Следует использовать только меркеры (M), глобальные DB или статическую память (в DB экземпляра) для назначения памяти M_BIT.

8.2 Принцип работы таймеров

С помощью таймерных инструкций можно устанавливаться программируемые задержки по времени. Количество таймеров для использования в программе пользователя ограничивается лишь объемом памяти CPU. Каждый таймер использует структуру DB размером 16 байт типа данных IEC_Timer для сохранения данных таймера, указываемых в верхней части инструкции. При добавлении инструкции STEP 7 автоматически создает DB.

Таблица 8-18 Таймерные инструкции

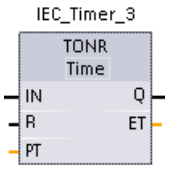
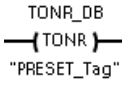
Блоки LAD/FBD	LAD катушки	SCL	Описание

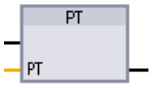
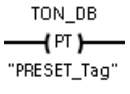
IEC_Timer_0
TP
Time
IN Q
PT ET
	TP_DB
	TP
	"PRESET_Tag"
		`"IEC_Timer_0_DB".TP(`
		`IN:=_bool_in_,`
		`PT:=_time_in_,`
		`Q=>_bool_out_,`
		`ET=>_time_out_);`
			Таймер TP генерирует импульс с заданной длительностью.

IEC_Timer_1
TON
Time
IN Q
PT ET
	TON_DB
	TON
	"PRESET_Tag"
		`"IEC_Timer_0_DB".TON (`
		`IN:=_bool_in_,`
		`PT:=_time_in_,`
		`Q=>_bool_out_,`
		`ET=>_time_out_);`
			Таймер TON устанавливает выход Q через заданное время на ВКЛ.

IEC_Timer_2
TOF
Time
IN Q
PT ET
	TOF_DB
	TOF
	"PRESET_Tag"
		`"IEC_Timer_0_DB".TOF (`
		`IN:=_bool_in_,`
		`PT:=_time_in_,`
		`Q=>_bool_out_,`
		`ET=>_time_out_);`
			Таймер TOF устанавливает выход Q через заданное время на ВЫКЛ.

Блоки LAD/FBD	LAD катушки	SCL	Описание

		`"IEC_Timer_0_DB".TONR (`
		`IN:=_bool_in_,`
		`R:=_bool_in_,`
		`PT:=_time_in_,`
		`Q=>_bool_out_,`
		`ET=>_time_out_);`
			Таймер TONR устанавливает выход Q через заданное время на ВКЛ. Истекшее время накапливается за несколько периодов тактирования до тех пор, пока вход R не будет запущен для сброса истекшего времени.

Только FBD:

		`PRESET_TIMER(`
		`PT:=_time_in_,`
		
		`TIMER:=_iec_timer_in_);`
			Катушка PT (заданное время) загружает новое ЗАДАННОЕ значение времени в указанный IEC_Timer.

Только FBD: 		`RESET_TIMER(` `_iec_timer_in_);`	Катушка RT (сбросить таймер) сбрасывает указанный IEC_Timer.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примерах SCL "IEC_Timer_0_DB" - это имя DB экземпляра.

Таблица 8-19 Типы данных для параметров

Параметр	Тип данных	Описание
Блок: IN Катушка: Поток сигналов	Bool	TP, TON и TONR: Блок: 0 = отключить таймер, 1 = включить таймер Катушка: Нет потока сигналов = отключить таймер, поток сигналов = включить таймер TOF: Блок: 0 = отключить таймер, 1 = включить таймер Катушка: Нет потока сигналов = включить таймер, поток сигналов = выключить таймер
R	Bool	Только TONR блок: 0 = не сбрасывать 1= сбросить истекшее время и Q бит на 0
Блок: PT Катушка: "PRESET_Tag"	Time	Блок таймера или катушка таймера: Вход заданного времени
Блок: Q Катушка: DBdata.Q	Bool	Блок таймера: Выход блока Q или бит Q в данных DB таймера Катушка таймера: Возможна только адресация бита Q в данных DB таймера
Блок: ET Катушка: DBdata.ET	Time	Блок таймера: Выход блока ET (истекшее время) или значение времени ET в данных DB таймера Катушка таймера: Возможна только адресация значения времени ET в данных DB таймера

Таблица 8-20 Реакция на изменение значений в параметрах PT и IN

Таймер	Изменения в параметрах блока PT и IN и соответствующих параметрах катушки
TP	• Изменение PT не влияет на работу, пока таймер запущен. • Изменение IN не влияет на работу, пока таймер запущен.
TON	• Изменение PT не влияет на работу, пока таймер запущен. • Изменение параметра IN на FALSE, пока таймер запущен, сбрасывает и останавливает таймер.
TOF	• Изменение PT не влияет на работу, пока таймер запущен. • Изменение параметра IN на TRUE, пока таймер запущен, сбрасывает и останавливает таймер.
TONR	• Изменение PT не оказывает влияния во время работы таймера. Но оказывает влияние, когда таймер возобновляет работу. • Изменение параметра IN на FALSE, пока таймер запущен, останавливает, но не сбрасывает таймер. Изменение IN обратно на TRUE запустит таймер с накопленного значения времени.

Значения PT (заданное время) и ET (истекшее время) хранятся в данных указанного DB IEC_TIMER, как двойные целые числа со знаком, представляющие значение времени в миллисекундах. Тип данных TIME использует идентификатор T# и может быть задан в простом (T#200ms или 200) или комбинированном формате времени, напр., T#2s_200ms.

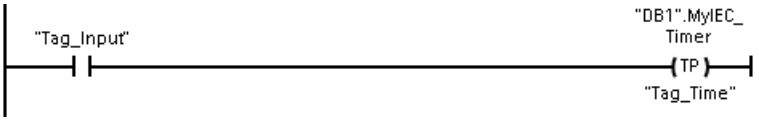
Таблица 8-21 Размер и диапазон типа данных TIME

Тип данных	Размер	Допустимый диапазон значений[1]
TIME	32 бит, хранится как данныеDInt	T#-24d_20h_31m_23s_648ms до T#24d_20h_31m_23s_647ms Сохраняется как: от -2.147.483.648 мс до +2.147.483.647 мс

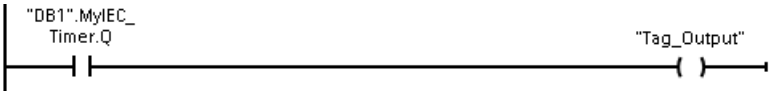
[1] Отрицательный диапазон представленного выше типа данных TIME не может использоваться для таймеров. Отрицательные значения для PT (заданное время) при исполнении таймерной инструкции сбрасываются на ноль. ET (истекшее время) всегда имеет положительное значение.

Пример для катушки таймера

Катушки таймера -(TP)-, -(TON)-, -(TOF)- и -(TONR)- должны быть последними инструкциями в сегменте LAD. Как показано в примере с таймером, инструкция контакта в следующем сегменте оценивает бит Q в данных DB IEC_Timer катушки таймера. Также следует адресовать элемент ELAPSED в данных DB IEC_Timer, если необходимо использовать значение истекшего времени в своей программе.



Импульсный таймер начинает свою работу при переходе из 0 в 1 значения бита в Tag_Input. Таймер запускается на время, определенное значением Tag_Time.



Пока таймер работает, состояние DB1.MyIEC_Timer.Q=1 и Tag_Output=1. По истечении значения для Tag_Time, DB1.MyIEC_Timer.Q=0 и Tag_Output=0.

Катушки сброса таймера -(RT)- предустановки таймера -(PT)-

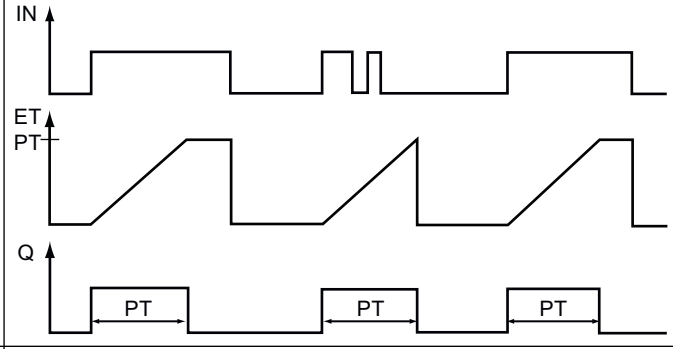
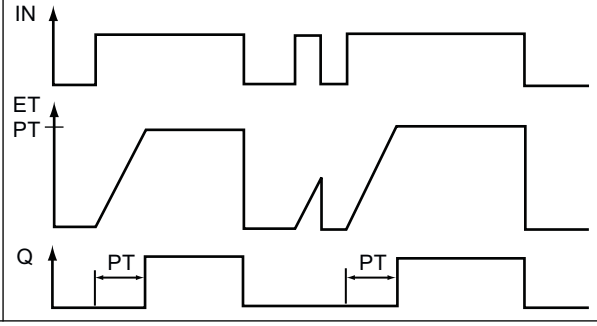
Эти инструкции катушек могут использоваться с таймерами-блоками или таймерами-катушками и могут быть помещены в промежуточную позицию Состояние сигнала на выходе катушки всегда соответствует состоянию сигнала на её входе. Когда катушка - (RT) - активируется, элемент времени ELAPSED в данных указанного DB IEC_Timer сбрасывается на 0. Когда катушка - (PT) - активируется, в элемент времени PRESET в данных указанного DB IEC_Timer загружается назначенное значение интервала времени.

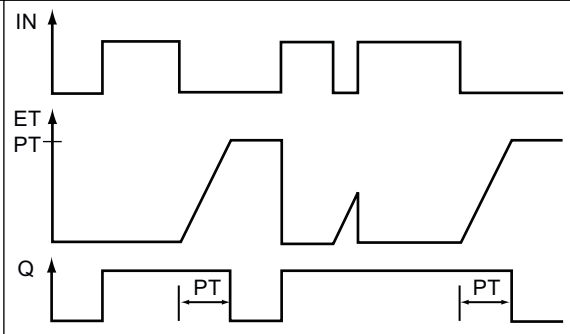
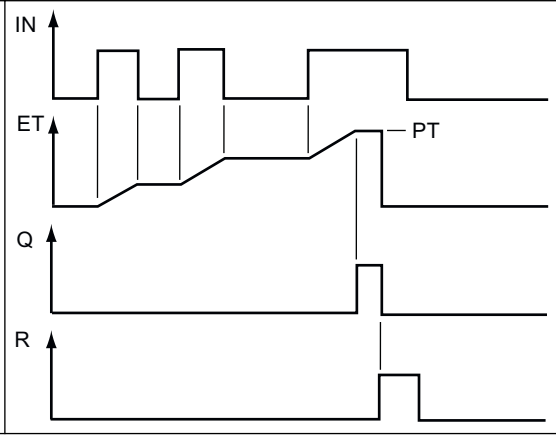
Примечание

Если таймеры помещаются в FB, то можно выбрать опцию "Многоэкземплярный блок данных". Имена структур таймеров могут отличаться у разных структур данных, но данные таймера находятся в одном единственном блоке данных и не требуют отдельного блока данных для каждого таймера. Это сокращает время обработки и объем памяти, необходимый для обработки таймеров Не существует каких-либо взаимодействий между структурами данных таймеров в совместно используемом мультиэкземплярном DB.

Принцип работы таймеров

Таблица 8-22 Типы IEC таймеров

Таймер	Диаграмма времени
TP: Генерация импульса Таймер TP генерирует импульс с заданной длительностью.	
TON: Генерация задержки включения Таймер TON устанавливает выход Q через заданное время на ВКЛ.	

Таймер	Диаграмма времени
TOF: Генерация задержки выключения Таймер TOF сбрасывает выход Q через заданное время на ВЫКЛ.	
TONR: Накопитель времени Таймер TONR устанавливает выход Q через заданное время на ВКЛ. Истекшее время накапливается за несколько периодов тактирования до тех пор, пока вход R не будет запущен для сброса истекшего времени.	

Примечание

CPU не выделяет ресурсы под определенные таймеры. Вместо этого каждый таймер использует свою собственную структуру в памяти DB и непрерывно работающий внутренний таймер CPU для выполнения синхронизации.

Когда таймер запускается по изменению фронта на входе инструкции TP, TON, TOF или TONR, значение непрерывно работающего внутреннего таймера CPU копируется в элемент START структуры DB, выделенной для этого таймера. Это стартовое значение остается неизменным в то время, пока таймер продолжает работать, и используется позже каждый раз, когда таймер обновляется. Каждый раз, когда таймер запускается, новое стартовое значение загружается в структуру таймера из внутреннего таймера CPU.

Когда таймер обновляется, стартовое значение, описанное выше, вычитается из текущего значения внутреннего таймера CPU, чтобы определить истекшее время. Истекшее время затем сравнивается с предварительной установкой, чтобы определить состояние бита таймера Q. Элементы ELAPSED и Q после обновляются в структуре DB, выделенной для этого таймера. Обратить внимание, что истекшее время фиксируется на уровне предварительно установленного значения (таймер не продолжает накапливать истекшее время после того, как предварительная установка достигнута).

Обновление таймера выполняется только того, когда:

- Выполняется таймерная инструкция (TP, TON, TOF или TONR)
- Адресация элемента "ELAPSED" структуры таймера в DB выполняется напрямую из инструкции
- Адресация элемента "Q" структуры таймера в DB выполняется напрямую из инструкции

Программирование таймеров

При планировании и создании программы пользователя необходимо учитывать следующие особенности работы таймеров:

- Возможно несколько обновлений таймера в одном цикле. Таймер обновляется каждый раз при выполнении таймерной инструкции (TP, TON, TOF, TONR), и каждый раз, когда элемент ELAPSED или Q структуры таймера используется в качестве параметра другой исполняемой инструкции. Это является преимуществом, когда нужны самые свежие данные времени (по сути - это прямое считывание таймера). Но для использования непротиворечивых значений в цикле программы, следует разместит таймерную инструкцию перед всеми другими инструкциями, которым нужны эти значения, и использовать переменные (теги) выходов Q и ET таймерной инструкции вместо элементов ELAPSED и Q структура таймера в DB.
- Возможны циклы без обновления таймера. Можно запустить таймер в функции, а затем перестать вызывать эту функцию снова для одного или нескольких циклов. Если никакие другие инструкции, которые адресуют элементы ELAPSED или Q структуры таймера, не выполняются, то таймер не будет обновлен. Новое обновление не произойдет до тех пор, пока либо инструкция таймера не будет выполнена снова, либо какая-либо другая инструкция будет выполнена с использованием ELAPSED или Q из структуры таймера в качестве параметра.

- Хотя это и не типично, но можно назначить одну и ту же структуру таймера DB нескольким таймерным инструкциям. В общем и целом, чтобы избежать неожиданного взаимодействия, необходимо использовать только один таймер (TP, TON, TOF, TONR) на структуру таймера DB.
- Таймеры с автосбросом полезны для запуска действий, которые должны выполняться регулярно. Как правило, таймеры с автосбросом создаются путем помещения NC-контакта, который ссылается на бит таймера, перед таймерной инструкцией. Этот сегмент таймера обычно расположен над одним или несколькими зависимыми сегментами, которые используют бит таймера для запуска действий. Когда время таймера истекает (истекшее время достигает заданного значения), бит таймера включается на один цикл, что позволяет выполнить логику управляемых битом таймера, зависимых сегментов. При следующем выполнении сегмента таймера NC-контакт выключается, сбрасывая таймер и удаляя бит таймера. В следующем цикле NC-контакт включен, что приводит к перезапуску таймера. При создании таймеров с автосбросом, схожих с этим, не использовать элемент "Q" структуры таймера в DB в качестве параметра для NC- контакта перед таймерной инструкцией. Вместо этого следует использовать для этой цели переменную (тег), связанную с выходом "Q" таймерной инструкции. Причина, по которой следует избегать обращения к элементу Q структуры таймера DB, заключается в том, что это вызывает обновление таймера. Если таймер обновляется через NC-контакт, то контакт немедленно сбрасывает таймерную инструкцию. Выход Q таймерной инструкции не будет включен в течение одного цикла, и зависимые сегменты не будут выполняться.

Сохранение данных таймера после перехода RUN-STOP-RUN или перезапуска CPU

Если сеанс в рабочем состоянии RUN завершается рабочим состоянием STOP или перезапуском CPU, а новый сеанс начинается в рабочем состоянии RUN, то данные таймера, сохраненные в предыдущем сеансе RUN, теряются, если только структура данных таймера не определена как структура с сохранением значений (таймеры TP, TON, TOF и TONR).

Если при вставке таймерной инструкции в редакторе текстов программ в диалоговом окне опций вызова принимаются значения по умолчанию, то автоматически назначается DB экземпляра, который **не может быть определен в качестве сохраняемого**. Для определения данных таймера как сохраняемых, следует использовать глобальный DB или мультиэкземплярный DB.

Энергонезависимое сохранение данных таймера путем назначения глобального DB

Данный способ работает независимо от того, где находится таймер (OB, FC или FB)

1. Создать глобальный DB:
   - Дважды кликнуть в дереве проекта по "Добавить новый блок".
   - Кликнуть по значку блока данных (DB).
   - Выбрать глобальный DB в качестве типа.
   - Если необходимо определить отдельные элементы данных в DB как сохраняемые, необходимо установить флажок "С оптимизацией" для типа DB. Другая опция для типа DB "По умолчанию - совместимость с S7 - 300/400" позволяет сделать сохраняемыми либо все элементы данных DB, либо никакие из них.
   - Нажать "ОК".
2. Добавление структур(ы) таймера к DB:
   - Добавить в новом глобальном DB новую статическую переменную (тег) с типом данных IEC_Timer.
   - Установить флажок в столбце "С сохранением", чтобы эта структура стала сохраняемой.
   - Повторить данную процедуру для создания структур для всех таймеров, которые необходимо сохранить в этом DB. Можно поместить каждую структуру таймера в уникальный глобальный DB, либо поместить несколько структур таймеров в один глобальный DB. Помимо таймеров в данный глобальный DB можно поместить и другие статический переменные (теги). Добавление нескольких структур таймеров в один глобальный DB снизит общее количество используемых блоков.
   - При необходимости переименовать структуры таймеров.
3. Открыть для редактирования программный блок (OB, FC или FB), в который будет вставлен сохраняемый таймер.
4. Разместить таймерную инструкцию в нужном месте.
5. При появлении диалогового окна с выбором вариантов вызова, нажать "Отмена".
6. Ввести вверху в новой таймерной инструкции имя глобального DB и структуры таймера, созданной ранее (не выбирать имя). Пример: "Data_block_3.Static_1".

Энергонезависимое сохранение данных таймера путем назначения многоэкземплярного DB

Данная возможность работает только при сохранении таймера в FB

Этот вариант зависит от того, установлена ли в свойствах FB опция "Оптимизированный доступ к блоку" (возможна только символьная адресация). С целью проверки конфигурации атрибута доступа для имеющегося FB, кликнуть правой кнопкой мыши в дереве проекта по FB и выбрать "Свойства", а потом "Атрибуты".

Если для FB установлен "Оптимизированный доступ к блоку" (возможна только символьная адресация):

1. Открыть FB для редактирования.
2. Разместить таймерную инструкцию в нужном месте FB.

3. При появлении диалогового окна с выбором вариантов вызова, кликнуть по символу мультиэкземпляра. Опция мультиэкземпляра доступна только при вставке инструкции в FB.
4. При необходимости переименовать таймер в появившемся окне вариантов вызова.
5. Нажать "ОК". Таймерная инструкция появляется в редакторе и структура IEC_TIMER появляется на интерфейсе FB в разделе "Статические".
6. При необходимости открыть редактор интерфейса FB (может потребоваться клик по небольшой стрелке для раскрытия представления).
7. Найти в "Статические" только что созданную структуру таймера.
8. Изменить выбор в столбце "С сохранением" этой структуры таймера на "С сохранением". Всякий раз, когда данный FB будет вызываться в другом программном блоке, будет создаваться DB экземпляра с интерфейсом, в котором структура таймера определена как "сохраняющаяся".

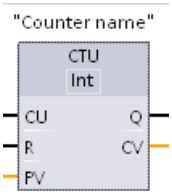
Если опция "Оптимизированный доступ к блоку" не указана для FB, то используется доступ к блоку по умолчанию, который совместим с классическими конфигурациями S7-300 / 400 и разрешает как символьный, так и прямой доступ. Чтобы назначить мультиэкземпляр для FB с доступом к блоку по умолчанию, выполнить следующие действия:

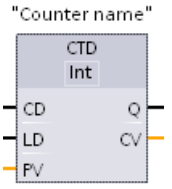
1. Открыть FB для редактирования.
2. Разместить таймерную инструкцию в нужном месте FB.
3. При появлении диалогового окна с выбором вариантов вызова, кликнуть по символу мультиэкземпляра. Опция мультиэкземпляра доступна только при вставке инструкции в FB.
4. При необходимости переименовать таймер в появившемся окне вариантов вызова.
5. Нажать "ОК". Таймерная инструкция появляется в редакторе и структура IEC_TIMER появляется на интерфейсе FB в разделе "Статические".
6. Открыть блок, который должен использовать этот FB.
7. Разместить это FB в нужном месте. Следствием является создание блока данных экземпляра для этого FB.
8. Открыть блок данных экземпляра, созданный при помещении FB в редактор.
9. Найти в "Статические" требуемую структуру таймера. Установить флажок в столбце "С сохранением" этой структуры таймера, чтобы определить эту структуру как "сохраняемую".

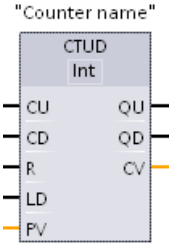
8.3 Принцип работы счётчиков

Таблица 8-23 Счетные инструкции

LAD/FBD	SCL	Описание

	`"IEC_Counter_0_DB".CTU(`
	`CU:=_bool_in,`
	`R:=_bool_in,`
	`PV:=_in,`
	`Q=>_bool_out,`
	`CV=>_out);`
		С помощь счетных инструкций можно подсчитывать внутренние события в программе и внешние события процесса. Каждый счетчик использует структуру, которая находится в блоке данных, чтобы хранить данные счетчика. При добавлении инструкции счетчика в редакторе, необходимо назначить блок данных.
		• CTU считает в прямом направлении (по возрастанию).
		• CTD считает в обратном направлении (по убыванию).
		• CTUD выполняет прямой и обратный счёт.

	`"IEC_Counter_0_DB".CTD(`
	`CD:=_bool_in,`
	`LD:=_bool_in,`
	`PV:=_in,`
	`Q=>_bool_out,`
	`CV=>_out);`	

	`"IEC_Counter_0_DB".CTUD(`
	`CU:=_bool_in,`
	`CD:=_bool_in,`
	`R:=_bool_in,`
	`LD:=_bool_in,`
	`PV:=_in_,`
	`QU=>_bool_out,`
	`QD=>_bool_out,`
	`CV=>_out_);`	

[1] В LAD и FBD: Выбрать тип данных для подсчитанных значений из выпадающего списка под именем инструкции.

[2] При добавлении инструкции, STEP 7 автоматически создает DB.

[3] В примерах SCL "IEC_Counter_0_DB" - это имя DB экземпляра.

Таблица 8-24 Типы данных для параметров

Параметр	Тип данных[1]	Описание
CU, CD	Bool	Прямой или обратный счет на одну единицу соответственно
R (CTU, CTUD)	Bool	Сброс состояния счетчика на ноль
LD (CTD, CTUD)	Bool	Управление загрузкой для предустановленного значения
PV	SInt, Int, DInt, USInt, UInt, UDInt	Предварительно установленное состояние счетчика
Q, QU	Bool	True, если CV >= PV

Параметр	Тип данных[1]	Описание
QD	Bool	True, если CV <= 0
CV	SInt, Int, DInt, USInt, UInt, UDInt	Текущее состояние счетчика

[1] Числовой диапазон подсчитанных значений зависит от выбранного типа данных. Если подсчитанное значение – целое без знака, то можно считать в обратном направлении до нуля, а в прямом направлении до границы диапазона. Если подсчитанное значение – целое со знаком, то можно считать в обратном направлении до нижнего предельного значения, а в прямом направлении - до верхнего предельного значения.

Количество счётчиков для использования в программе пользователя ограничивается лишь объемом памяти CPU. Необходимый для счётчиков объем памяти:

- Для типов данных SInt или USInt счетная инструкция использует 3 байта.
- Для типов данных Int или UInt счетная инструкция использует 6 байтов.
- Для типов данных DInt или UDInt счетная инструкция использует 12 байт.

Эти инструкции используют программные счетчики, максимальная скорость счета которых ограничена частотой вызовов OB, в которых они находятся. OB, в который вставляются инструкции, должен выполняться достаточно часто, чтобы обнаружить все переходы входов CU или CD. Для ускорения счета следует использовать инструкцию CTRL_HSC (Страница 557).

Примечание

Если счетные инструкции помещаются в FB, то можно выбрать опцию "Многоэкземплярный блок данных". Имена структур счетчиков могут отличаться у разных структур данных, но данные счетчика находятся в одном единственном блоке данных и не требуют отдельного блока данных для каждого счетчика. Это сокращает время обработки и объем памяти, необходимый для обработки счетчиков. Не существует каких-либо взаимодействий между структурами данных счетчиков в совместно используемом мультиэкземплярном DB.

Принцип работы счётчиков

Таблица 8-25 Принцип работы CTU (прямой счёт)

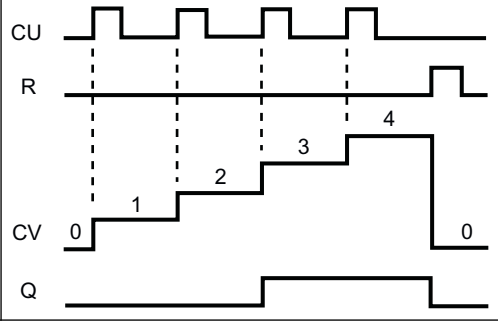
Счетчик	Принцип
Счетчик CTU увеличивает значение на 1 каждый раз, когда значение параметра CU изменяется из 0 на 1. На временной диаграмме CTU показан принцип работы для целочисленного значения счета без знака (где PV = 3). • Если значение параметра CV (текущее значение счетчика) больше или равно значению параметра PV (заданное состояние счетчика), тогда параметр для выхода счетчика Q = 1. • Если значение параметра сброса R изменяется с 0 на 1, то текущее значение счетчика сбрасывается на 0.	

Таблица 8-26 Принцип работы CTD (обратный счёт)

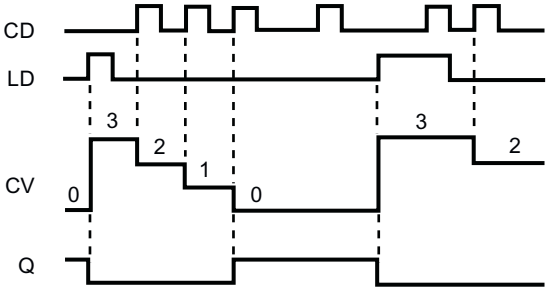
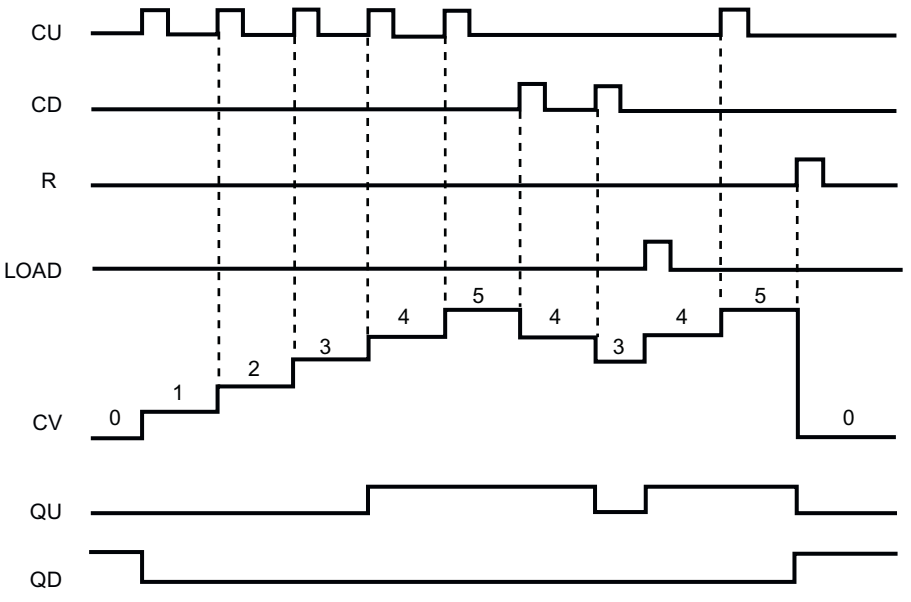
Счетчик	**Принцип**
Счетчик CTD уменьшает значение на 1 каждый раз, когда значение параметра CD изменяется из 0 на 1. На временной диаграмме CTD показан принцип работы для целочисленного подсчитанного значения без знака (где PV = 3). • Если значение параметра CV (текущее значение счетчика) меньше или 0, тогда параметр для выхода счетчика Q = 1. • Если значение параметра LOAD изменяется с 0 на 1, то значение на параметре PV (заданное значение) загружается как новое CV (текущее значение счётчика) в счетчик.	

Таблица 8-27 Принцип работы CTUD (прямой и обратный счёт)

Счетчик	**Принцип**
Счетчик CTUD увеличивает или уменьшает значение на 1 каждый раз, когда вход прямого счета (CU) или вход обратного счета (CD) изменяется из 0 на 1. На временной диаграмме CTUD показан принцип работы для целочисленного подсчитанного значения без знака (где PV = 4). • Если значение параметра CV больше или равно значению параметра PV, тогда параметр для выхода счетчика QU = 1. • Если значение параметра CV меньше или 0, тогда параметр для выхода счетчика QD = 1. • Если значение параметра LOAD изменяется с 0 на 1, то значение на параметре PV загружается как новое CV в счетчик. • Если значение параметра сброса R изменяется с 0 на 1, то текущее значение счетчика сбрасывается на 0.	

Сохранение данных счетчика после перехода RUN-STOP-RUN или перезапуска CPU

Если сеанс в рабочем состоянии RUN завершается рабочим состоянием STOP или перезапуском CPU, а новый сеанс начинается в рабочем состоянии RUN, то данные счетчика, сохраненные в предыдущем сеансе RUN, теряются, если только структура данных счетчика не определена как структура с сохранением значений (счетчики TP, TON, TOF и TONR).

Если при вставке счетной инструкции в редакторе текстов программ в диалоговом окне опций вызова принимаются значения по умолчанию, то автоматически назначается DB экземпляра, который **не может быть определен в качестве сохраняемого**. Для определения данных счетчика как сохраняемых, следует использовать глобальный DB или мультиэкземплярный DB.

Энергонезависимое сохранение данных счетчика путем назначения глобального DB

Данный способ работает независимо от того, где находится счетчик (OB, FC или FB)

1. Создать глобальный DB:
   - Дважды кликнуть в дереве проекта по "Добавить новый блок".
   - Кликнуть по значку блока данных (DB).
   - Выбрать глобальный DB в качестве типа.
   - Если необходимо определить отдельные элементы данных в DB как сохраняемые, необходимо установить флажок "Только символьная адресация".
   - Нажать "OK".
2. Добавление структур(ы) счетчика к DB:
   - Добавить в новом глобальном DB новую статическую переменную (тег) с типом данных "Счетчик". При этом следует учитывать предустановленный тип и значение счета.
   - Установить флажок в столбце "С сохранением", чтобы эта структура стала сохраняемой.
   - Повторить данную процедуру для создания структур для всех счетчиков, которые необходимо сохранить в этом DB. Можно поместить каждую структуру счетчика в уникальный глобальный DB, либо поместить несколько структур счетчиков в один глобальный DB. Помимо счетчиков в данный глобальный DB можно поместить и другие статический переменные (теги). Добавление нескольких структур счетчиков в один глобальный DB снизит общее количество используемых блоков.
   - При необходимости переименовать структуры счетчиков.
3. Открыть для редактирования программный блок (OB, FC или FB), в который будет вставлен сохраняемый счетчик.
4. Разместить счетную инструкцию в нужном месте.

5. При появлении диалогового окна с выбором вариантов вызова, нажать "Отмена". Теперь должна появиться новая счетная инструкция с замещением "???" над и под именем инструкции.
6. Ввести вверху в новой счетной инструкции имя глобального DB и структуры счетчика, созданной ранее (не выбирать имя). Пример: "Data_block_3.Static_1". После задается соответствующий тип для предустановки и значение счета (пример: UInt для структуры IEC_UCounter).

Тип данных счетчика	Соответствующий тип для предустановки и значение счета
IEC_Counter	INT
IEC_SCounter	SINT
IEC_DCounter	DINT
IEC_UCounter	UINT
IEC_USCounter	USINT
IEC_UDCounter	UDINT

Энергонезависимое сохранение данных счетчика путем назначения многоэкземплярного DB

Данная возможность работает только при сохранении счетчика в FB.

Этот вариант зависит от того, установлена ли в свойствах FB опция "Оптимизированный доступ к блоку" (возможна только символьная адресация). С целью проверки конфигурации атрибута доступа для имеющегося FB, кликнуть правой кнопкой мыши в дереве проекта по FB и выбрать "Свойства", а потом "Атрибуты".

Если для FB установлен "Оптимизированный доступ к блоку" (возможна только символьная адресация):

1. Открыть FB для редактирования.
2. Разместить счетную инструкцию на нужном месте в FB.
3. При появлении диалогового окна с выбором вариантов вызова, кликнуть по символу мультиэкземпляра. Опция мультиэкземпляра доступна только при вставке инструкции в FB.
4. При необходимости присвоит счетчику имя в окне вариантов вызова.
5. Нажать "ОК". Счетная инструкция появляется в редакторе с типом INT для предустановки и значения счета, а структура IEC_COUNTER появляется на интерфейсе FB в разделе "Статические".
6. При необходимости изменить тип в счетной инструкции с INT на другой тип. Структура счетчика соответственно изменится.
7. При необходимости открыть редактор интерфейса FB (может потребоваться клик по небольшой стрелке для раскрытия представления).

8. Найти в "Статические" только что созданную структуру счетчика.
9. Изменить выбор в столбце "С сохранением" этой структуры счетчика на "С сохранением". Всякий раз, когда данный FB будет вызываться в другом программном блоке, будет создаваться DB экземпляра с интерфейсом, в котором структура счетчика определена как "сохраняющаяся".

Если опция "Оптимизированный доступ к блоку" не указана для FB, то используется доступ к блоку по умолчанию, который совместим с классическими конфигурациями S7-300 / 400 и разрешает как символьный, так и прямой доступ. Чтобы назначить мультиэкземпляр для FB с доступом к блоку по умолчанию, выполнить следующие действия:

1. Открыть FB для редактирования.
2. Разместить счетную инструкцию на нужном месте в FB.
3. При появлении диалогового окна с выбором вариантов вызова, кликнуть по символу мультиэкземпляра. Опция мультиэкземпляра доступна только при вставке инструкции в FB.
4. При необходимости присвоит счетчику имя в окне вариантов вызова.
5. Нажать "ОК". Счетная инструкция появляется в редакторе с типом INT для предустановки и значения счета, а структура IEC_COUNTER появляется на интерфейсе FB в разделе "Статические".
6. При необходимости изменить тип в счетной инструкции с INT на другой тип. Структура счетчика соответственно изменится.
7. Открыть блок, который должен использовать этот FB.
8. Разместить это FB в нужном месте. Следствием является создание блока данных экземпляра для этого FB.
9. Открыть блок данных экземпляра, созданный при помещении FB в редактор.
10. Найти в "Статические" требуемую структуру счетчика. Установить флажок в столбце "С сохранением" этой структуры счетчика, чтобы определить эту структуру как "сохраняемую".

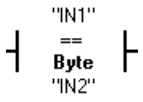
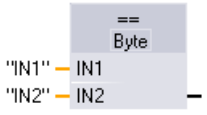
Тип в счетной инструкции (для предустановки и значения счета)	Соответствующий тип структуры на интерфейсе FB
INT	IEC_Counter
SINT	IEC_SCounter
DINT	IEC_DCounter
UINT	IEC_UCounter
USINT	IEC_USCounter
UDINT	IEC_UDCounter

8.4 Принцип работы компараторов

8.4.1 Операции сравнения

Таблица 8-28 Операции сравнения

LAD	FBD	SCL	Описание

		`out := in1 = in2;`
		or
		`IF in1 = in2`
		`  THEN out := 1;`
		`  ELSE out := 0;`
		`  END_IF;`
			Сравнение двух значений с одним типом данных. Если сравнение на LAD контакте равно TRUE, то контакт активируется. Если сравнение в FBD блоке TRUE, тогда выход блока также TRUE.

[1] В LAD и FBD: Кликнуть по имени инструкции (таком как "=="), чтобы выбрать тип сравнения из выпадающего списка. Нажать на "???" и выбрать тип данных из выпадающего списка.

Таблица 8-29 Типы данных для параметров

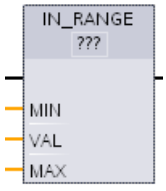
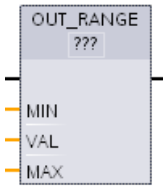
Параметр	Тип данных	Описание
IN1, IN2	Byte, Word, DWord, SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, String, WString, Char, Char, Time, Date, TOD, DTL, Constant	Сравниваемые значения

Таблица 8-30 Описания функций сравнения

Тип отношения	Сравнения ИСТИННО, если...
=	IN1 равен IN2
<>	IN1 не равен IN2
>=	IN1 больше или равен IN2
<=	IN1 меньше или равен IN2
>	IN1 больше чем IN2
<	IN1 меньше чем IN2

8.4.2 Инструкции IN_Range (значение в диапазоне) и OUT_Range (значение вне диапазона)

Таблица 8-31 Инструкции проверки нахождения значения в пределах диапазона и вне диапазона

LAD/FBD	SCL	Описание
	`out := IN_RANGE(min, val, max);`	Проверяет, входит ли входное значение в указанный диапазон значений или нет. Если сравнение TRUE, то выход блока TRUE
	`out := OUT_RANGE(min, val, max);`	

[1] В LAD и FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.

Таблица 8-32 Типы данных для параметров

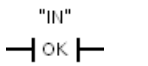
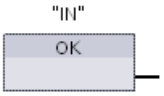
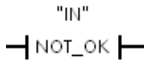
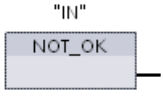
Параметр	Тип данных[1]	Описание
MIN, VAL, MAX	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Constant	Входы компаратора

[1] Входные параметры MIN, VAL, и MAX должны иметь один тип данных.

- Сравнение IN_RANGE ИСТИННО, если: MIN <= VAL <= MAX
- Сравнение OUT_RANGE ИСТИННО, если: VAL < MIN или VAL > MAX

8.4.3 Инструкции OK (проверить на достоверность) и NOT_OK (проверить на недостоверность)

Таблица 8-33 Инструкции OK (проверить на достоверность) и NOT_OK (проверить на недостоверность)

LAD	FBD	SCL	Описание
		Недоступно	Проверяет вещественное значение на входе на соответствие IEEE спецификации 754
		Недоступно	

[1] В LAD и FBD: Если контакт в LAD принимает значение ИСТИНА, то он активируется и пропускает через себя поток сигналов. Если блок FBD принимает значение ИСТИНА, то выход блока тоже принимает значение ИСТИНА.

Таблица 8-34 Типы данных для параметров

Параметр	Тип данных	Описание
IN	Real, LReal	Входные данные

Таблица 8-35 Принцип работы

Инструкция	Проверка, является ли число вещественным, принимает значение ИСТИНА, если:
OK	Входное значение является действительным вещественным числом[1].
NOT_OK	Входное значение не является действительным вещественным числом[1].

[1] Значение типа Real или LReal является недостоверным, если оно равно +/- INF (бесконечность), NaN (Not a Number, не число), или если оно является денормализованным числом. Денормализованное число – это число, очень близкое к нулю. При расчетах CPU вместо денормализованного числа подставляет ноль.

8.4.4 Операции сравнения типов данных Variant и Array

8.4.4.1 Операции сравнения по равенству и неравенству

S7-1200 CPU предлагает инструкции для опроса типа данных переменной (тега), на которую указывает операнд Variant, и для сравнения, совпадает ли она с типом данных другого операнда (равна) или нет (не равна).

Кроме этого, S7-1200 CPU предлагает инструкции для опроса типа данных элемента массива на предмет равенства или неравенства типу данных другого операнда.

В этих операциях <Operand1> сравнивается с <Operand2>. <Operand1> должен иметь тип данных Variant. <Operand2> может быть элементарным типом данных PLC. В LAD и FBD <Operand1> - это операнд над инструкцией. В LAD <Operand2> - это операнд под инструкцией.

Для всех инструкций результат логической операции (RLO) равен 1 (истина), если проверка на равенство или неравенство прошла успешно, и 0 (ложь) если нет.

Существуют следующие инструкции сравнения на равенство и неравенство:

- EQ_Type (сравнение типа данных с типом данных переменной (тега) на предмет РАВЕНСТВА)
- NE_Type (сравнение типа данных с типом данных переменной (тега) на предмет НЕРАВЕНСТВА)
- EQ_ElemType (сравнение типа данных элемента МАССИВА с типом данных переменной (тега) на предмет РАВЕНСТВА)
- NE_ElemType (сравнение типа данных элемента МАССИВА с типом данных переменной (тега) на предмет НЕРАВЕНСТВА)

Таблица 8-36 Инструкции EQ и NE

LAD	FBD	SCL	Описание
#Operand1 EQ_Type "Operand2"	#Operand1 EQ_Type "Operand2" — IN2 OUT	Недоступно	Проверяет, имеет ли переменная (тег), на который указывает Variant в Operand1, тот же тип данных, что и переменная (тег) в Operand2.
#Operand1 NE_Type "Operand2"	#Operand1 NE_Type "Operand2" — IN2 OUT	Недоступно	Проверяет, имеет ли переменная (тег), на который указывает Variant в Operand1, тип данных, отличный от типа данных переменной (тега) в Operand2.
#Operand1 EQ_ElemType "Operand2"	#Operand1 EQ_ElemType "Operand2" — IN2 OUT	Недоступно	Проверяет, имеет ли элемент массива, на который указывает Variant в Operand1, тот же тип данных, что и переменная (тег) в Operand2.
#Operand1 NE_ElemType "Operand2"	#Operand1 NE_ElemType "Operand2" — IN2 OUT	Недоступно	Проверяет, имеет ли элемент массива, на который указывает Variant в Operand1, тип данных, отличный от типа данных переменной (тега) в Operand2.

Таблица 8-37 Типы данных для параметров

Параметр	Тип данных	Описание
Operand1	Variant	Первый операнд
Operand2	Битовые строки, целые числа, числа с плавающей запятой, таймеры, дата и время, символьные строки, МАССИВ, типы данных PLC	Второй операнд

8.4.4.2 Операции сравнения с нулевым значением

Можно использовать инструкции IS_NULL и NOT_NULL для определения, указывает ли вход на объект или нет.

Для обоих инструкций <Operand> должен иметь тип данных Variant.

Таблица 8-38 IS_NULL (запрос на РАВЕНСТВО указателя НУЛЮ) и NOT_NULL (запрос на ОТЛИЧИЕ указателя от НУЛЯ)

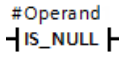
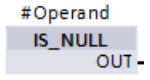
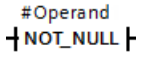
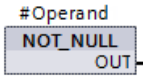
LAD	FBD	SCL	Описание
		Недоступно	Проверяет, является ли переменная (тег), на которую указывает элемент Variant в Operand, нулем и тем самым не является объектом.
		Недоступно	Проверяет, отличается ли переменная (тег), на которую указывает элемент Variant в Operand, от нуля и тем самым является объектом.

Таблица 8-39 Типы данных для параметров

Параметр	Тип данных	Описание
Operand	Variant	Операнд для проверки на предмет равенства или неравенства нулю.

8.4.4.3 IS_ARRAY (проверка на МАССИВ)

С помощью инструкции "Проверить на МАССИВ" можно запросить, указывает ли элемент Variant на переменную (тег) типа данных Array.

<операнд> должен иметь тип данных Variant.

Инструкция возвращает 1 (истина), если операнд является массивом.

Таблица 8-40 IS_ARRAY (проверка на МАССИВ)

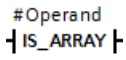
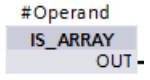
LAD	FBD	SCL	Описание
		IS_ARRAY(_variant_in_)	Проверяет, является ли переменная (тег), на которую указывает элемент Variant в Operand массивом.

Таблица 8-41 Типы данных для параметров

Параметр	Тип данных	Описание
Operand	Variant	Операнд для проверки, идет ли речь о массиве.

8.5 Арифметические функции

8.5.1 CALCULATE (вычисление)

Таблица 8-42 Инструкция CALCULATE

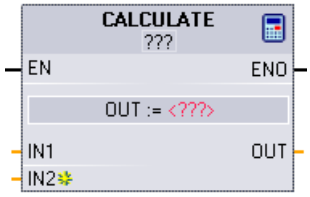
LAD/FBD	SCL	Описание
	Использование обычных математических SCL выражений для создания уравнения.	Инструкция CALCULATE позволяет создать математическую функцию, которая обрабатывает входы (IN1, IN2, .. INn) и водит результат на OUT согласно заданному пользователем уравнению. • Сначала необходимо выбрать тип данных. Все входы и выход должны иметь один и тот же тип данных. • Для добавления нового входа, кликнуть по символу на последнем входе.

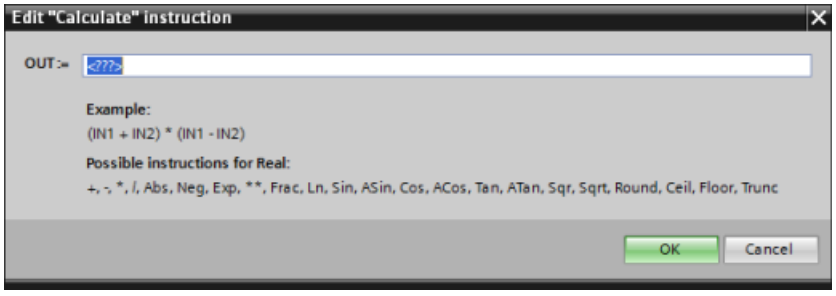
Таблица 8-43 Типы данных для параметров

Параметр	Тип данных[1]
IN1, IN2, ..INn	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord

[1] Параметры IN и OUT должны иметь одинаковый тип данных (с неявными преобразованиями входных параметров). Пример: Значение с типом SINT для входа будет преобразовано в INT или REAL, если OUT имеет тип INT или REAL соответственно.

Кликнуть по значку калькулятора, чтобы открыть диалоговое окно и определить математическую функцию. Уравнение вводится как входы (как IN1 и IN2) и операции. После нажатия "OK" для сохранения функции, диалоговое окно автоматически создает входы для инструкции CALCULATE.

Диалоговое окно отображает пример и список возможных инструкций, которые можно использовать на основе типа данных параметра OUT:



Примечание

Также необходимо создать вход для констант в функции. Постоянное значение после вводится на назначенном входе инструкции CALCULATE.

Назначив константы в качестве входов, можно скопировать инструкцию CALCULATE в другие места программы пользователя без необходимости изменения функции. После этого можно изменить значения или переменные (теги) входов для инструкции без модификации самой функции.

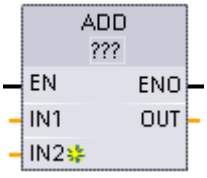
При выполнении инструкции CALCULATE и успешном завершении всех отдельных операций во время вычисления, ENO = 1. В ином случае ENO = 0.

Пример инструкции CALCULATE можно найти в "Создание сложного уравнения с помощью простой инструкции (Страница 41)".

8.5.2 Инструкции сложения, вычитания, умножения и деления

Таблица 8-44 Инструкции сложения, вычитания, умножения и деления

LAD/FBD	SCL	Описание

	`out := in1 + in2;`
	`out := in1 - in2;`
	`out := in1 * in2;`
	`out := in1 / in2;`
		• ADD: Сложение (IN1 + IN2 = OUT)
		• SUB: Вычитание (IN1 - IN2 = OUT)
		• MUL: Умножение (IN1 * IN2 = OUT)
		• DIV: Деление (IN1 / IN2 = OUT)
		Операция целочисленного деления отбрасывает дробную часть частного, чтобы получить целое число на выходе.

[1] В LAD и FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.

Таблица 8-45 Типы данных для параметров (LAD и FBD)

Параметр	Тип данных[1]	Описание
IN1, IN2	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Constant	Входы арифметической операции
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Выход арифметической операции

[1] Параметры IN1, IN2 и OUT должны иметь одинаковый тип данных.



Для добавления входа ADD или MUL кликнуть левой кнопкой мыши по символу "Создать" или правой кнопкой мыши на входном соединении одного из параметров IN и выбрать команду "Вставить вход".

Для удаления входа кликнуть правой кнопкой мыши на входном соединении одного из параметров IN (если имеется более двух исходных входов) и выбрать команду "Удалить".

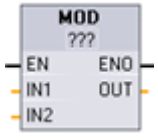
При активации (EN = 1) арифметическая инструкция выполняет указанную функцию для входных значений (IN1 и IN2) и сохраняет результат в ячейке памяти, заданной выходным параметром (OUT). После успешного выполнения операции, ENO= 1.

Таблица 8-46 Состояние ENO

ENO	Описание
1	Ошибки отсутствуют
0	Результирующе значение арифметической операции находится вне допустимого диапазона значений для выбранного типа данных. Возвращается наименьшая значимая часть результата, соответствующая целевому размеру.
0	Деление на 0 (IN2 = 0): Результат не определен, и возвращается ноль.
0	Real/LReal: Если одно из значений представляет собой NaN (not a number, не число), тогда возвращается значение NaN.
0	ADD Real/LReal: Если оба входных значения равны бесконечности (INF) с разными знаками, то эта операция недопустима, и в качестве результата возвращается NaN.
0	SUB Real/LReal: Если оба входных значения равны бесконечности (INF) с одним знаком, то эта операция недопустима, и в качестве результата возвращается NaN.
0	MUL Real/LReal: Если одно из входных значений равно нулю, а другое INF, то эта операция недопустима и в качестве результата возвращается NaN.
0	DIV Real/LReal: Если оба входных значения равны нулю или INF, то эта операция недопустима и в качестве результата возвращается NaN.

8.5.3 MOD (остаток от деления)

Таблица 8-47 MOD (остаток от деления)

LAD/FBD	SCL	Описание
	`out := in1 MOD in2;`	Можно использовать инструкцию MOD, чтобы вывести остаток от операции целочисленного деления. Значение на входе IN1 делится на значение на входе IN2, а остаток выводится на выходе OUT.

[1] В LAD и FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.

Таблица 8-48 Типы данных для параметров

Параметр	Тип данных[1]	Описание
IN1 и IN2	SInt, Int, DInt, USInt, UInt, UDInt, Constant	Входы инструкции получения остатка
OUT	SInt, Int, DInt, USInt, UInt, UDInt	Выход инструкции получения остатка

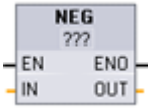
[1] Параметры IN1, IN2 и OUT должны иметь одинаковый тип данных.

Таблица 8-49 Значения ENO

ENO	Описание
1	Ошибки отсутствуют
0	Значение IN2 = 0, OUT будет присвоено значение ноль

8.5.4 NEG (создать двоичное дополнение)

Таблица 8-50 Инструкция NEG (создать двоичное дополнение)

LAD/FBD	SCL	Описание
	`-(in);`	Инструкция NEG инвертирует арифметический знак значения параметра IN и сохраняет результат в параметр OUT.

[1] В LAD и FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.

Таблица 8-51 Типы данных для параметров

Параметр	Тип данных[1]	Описание
IN	SInt, Int, DInt, Real, LReal, Constant	Вход арифметической операции
OUT	SInt, Int, DInt, Real, LReal	Выход арифметической операции

[1] Параметры IN и OUT должны иметь одинаковый тип данных.

Таблица 8-52 Состояние ENO

ENO	Описание
1	Ошибки отсутствуют
0	Результирующе значение находится вне допустимого диапазона значений для выбранного типа данных. Пример для SInt: NEG (-128) дает +128, что превышает максимальное значение для этого типа данных.

8.5.5 INC (увеличение на 1) и DEC (уменьшение на 1)

Таблица 8-53 Инструкции INC и DEC

LAD/FBD	SCL	Описание
	`in_out := in_out + 1;`	Увеличивает на единицу целое число со знаком или без знака: Значение IN_OUT +1 = Значение IN_OUT
	`in_out := in_out - 1;`	Уменьшает на единицу целое число со знаком или без знака: Значение IN_OUT - 1 = Значение IN_OUT

[1] В LAD и FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.

Таблица 8-54 Типы данных для параметров

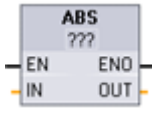
Параметр	Тип данных	Описание
IN/OUT	SInt, Int, DInt, USInt, UInt, UDInt	Вход и выход арифметической операции

Таблица 8-55 Состояние ENO

ENO	Описание
1	Ошибки отсутствуют
0	Результирующе значение находится вне допустимого диапазона значений для выбранного типа данных. Пример для SInt: INC (+127) дает +128, что превышает максимальное значение для этого типа данных.

8.5.6 ABS (создание абсолютного значения)

Таблица 8-56 Инструкция ABS (создание абсолютного значения)

LAD/FBD	SCL	Описание
	`out := ABS(in);`	Вычисляет абсолютное значение целого числа со знаком или вещественного числа на входе параметре IN и сохраняет результат в параметре OUT.

[1] В LAD и FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.

Таблица 8-57 Типы данных для параметров

Параметр	Тип данных[1]	Описание
IN	SInt, Int, DInt, Real, LReal	Вход арифметической операции
OUT	SInt, Int, DInt, Real, LReal	Выход арифметической операции

[1] Параметры IN и OUT должны иметь одинаковый тип данных.

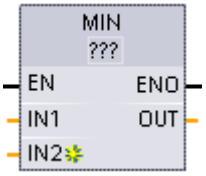
Таблица 8-58 Состояние ENO

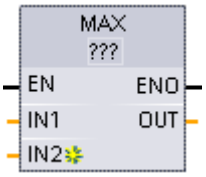
ENO	Описание
1	Ошибки отсутствуют
0	Результирующе значение арифметической операции находится вне допустимого диапазона значений для выбранного типа данных. Пример для SInt: ABS (-128) дает +128, что превышает максимальное значение для этого типа данных.

8.5.7 MIN (вычислить минимум) и MAX (вычислить максимум)

Таблица 8-59 Инструкции MIN (вычислить минимум) и MAX (вычислить максимум)

LAD/FBD	SCL	Описание

	`out:= MIN(`
	`in1:=_variant_in_,`
	`in2:=_variant_in_`
	`[,...in32]);`
		Инструкция MIN сравнивает значение двух параметров IN1 и IN2, после чего присваивает меньшее значение параметру OUT.

	`out:= MAX(`
	`in1:=_variant_in_,`
	`in2:=_variant_in_`
	`[,...in32]);`
		Инструкция MAX сравнивает значение двух параметров IN1 и IN2, после чего присваивает большее значение параметру OUT.

[1] В LAD и FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.

Таблица 8-60 Типы данных для параметров

Параметр	Тип данных[1]	Описание
IN1, IN2 [...IN32]	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD, Constant	Входы арифметической операции (макс. 32 входа)
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD	Выход арифметической операции

[1] Параметры IN1, IN2 и OUT должны иметь одинаковый тип данных.



Для добавления входа кликнуть по символу "Создать" или правой кнопкой мыши по одному из параметров IN на входном соединителе выбрать команду "Вставить вход".

Для удаления входа кликнуть правой кнопкой мыши на входном соединении одного из параметров IN (если имеется более двух исходных входов) и выбрать команду "Удалить".

Таблица 8-61 Состояние ENO

ENO	Описание
1	Ошибки отсутствуют
0	Только для типа данных Real: • По крайней мере один из входов не является вещественным (REAL) числом (NaN). • Результат на выходе OUT = +/- INF (бесконечность).

8.5.8 LIMIT (установка предельного значения)

Таблица 8-62 Инструкция LIMIT (установка предельного значения)

LAD/FBD	SCL	Описание

	`LIMIT(MN:=_variant_in_,`
	`IN:=_variant_in_,`
	`MX:=_variant_in_,`
	`OUT:=_variant_out_);`
		С помощью инструкции Limit можно проверить, лежит ли значение параметра IN в пределах заданного с помощью параметров MIN иMAX and if not, clamps the value at MIN or MAX. диапазона значений.

[1] В LAD и FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.

Таблица 8-63 Типы данных для параметров

Параметр	Тип данных[1]	Описание
MN, IN и MX	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD· Constant	Входы арифметической операции
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD	Выход арифметической операции

[1] Параметры MN, IN, MX и OUT должны иметь одинаковый тип данных.

Если значение параметра IN лежит в пределах указанного диапазона, то значение для IN сохраняется в параметр OUT. Если значение параметра IN лежит вне указанного диапазона, то в OUT выводится значение параметра MIN (если значение IN меньше, чем

значение MIN), или значение параметра MAX (если значение: IN больше, чем значение MAX).

Таблица 8-64 Состояние ENO

ENO	Описание
1	Ошибки отсутствуют
0	Real: Если одно или несколько значений MIN, IN или MAX представляют собой NaN (не число), тогда возвращается NaN.
0	Если MIN больше, чем MAX, то значение IN присваивается выходу OUT.

Примеры SCL:

- MyVal := LIMIT(MN:=10,IN:=53, MX:=40); //Результат: MyVal = 40
- MyVal := LIMIT(MN:=10,IN:=37, MX:=40); //Результат: MyVal = 37
- MyVal := LIMIT(MN:=10,IN:=8, MX:=40); //Результат: MyVal = 10

8.5.9 Экспоненциальные, логарифмические и тригонометрические инструкции

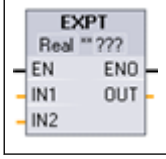
С помощью инструкций с плавающей запятой можно программировть арифметические функции с типом данных Real или LReal:

- SQR: Определение квадрата ($IN^2 = OUT$)
- SQRT: Определение квадратного корня ($\sqrt{IN} = OUT$)
- LN: Определение натурального логарифма (LN(IN = OUT)
- EXP: Определение экспоненциального значения ($e^{IN} = OUT$), по основанию e = 2,71828182845904523536
- EXPT: Возведение в степень ($IN1^{IN2} = OUT$)
  Параметры IN1 и OUT инструкции EXPT всегда имеют одинаковый тип данных, для которого необходимо выбрать Real или LReal. Тип данных для параметра экспоненты IN2 может быть выбран из нескольких типов данных.
- FRAC: Возврат мест после запятой (место после запятой числа с плавающей точкой IN = OUT)
- SIN: Определение значения синуса (sin(IN в радианах) = OUT)
- ASIN: Определение значения арксинуса (arcsine(IN) = OUT в радианах), где sin(OUT в радианах) = IN
- COS: Определение значения косинуса (cos(IN в радианах) = OUT)
- ACOS: Определение значения арккосинуса (arccos(IN) = OUT в радианах), где cos(OUT в радианах) = IN
- TAN: Определение значения тангенса (tan(IN radians) = OUT)
- ATAN: Определение значения арктангенса (arctan(IN в радианах), где tan(OUT в радианах) = IN

Таблица 8-65 Примеры арифметических инструкций для чисел с плавающей запятой

LAD/FBD	SCL	Описание

	`out := SQR(in);`
	или
	`out := in * in;`
		Квадрат: IN [2] = OUT
		Пример: Если IN = 9, тогдаOUT = 81.

	`out := in1 ** in2;`	Возведение в степень: IN1 ^{IN2} = OUT Пример: Если IN1 = 3 иIN2 = 2, тогдаOUT = 9.

[1] В LAD и FBD: Нажать на "???" (для имени инструкции) и выбрать тип данных из выпадающего списка.

[2] В SCL: Можно использовать обычные математические операторы для создания математических выражений в SCL.

Таблица 8-66 Типы данных для параметров

Параметр	Тип данных	Описание
IN, IN1	Real, LReal, Constant	Входы
IN2	SInt, Int, DInt, USInt, UInt,UDInt, Real, LReal, Constant	Вход показателя EXPT
OUT	Real, LReal	Выходы

Таблица 8-67 Состояние ENO

ENO	Инструкция	Условие	Результат (OUT)
1	Все	Ошибки отсутствуют	Правильный результат
0	SQR	Результат превышает допустимый диапазон для Real/LReal	+INF
		IN = +/- NaN (не является числом)	+NaN
	SQRT	IN отрицательный	-NaN
		IN = +/- INF (бесконечность) или +/- NaN	+/- INF или +/- NaN
	LN	IN = 0,0, отрицательный, -INF или -NaN	-NaN
		IN = +INF или +NaN	+INF или +NaN
	EXP	Результат превышает допустимый диапазон для Real/LReal	+INF
		IN = +/- NaN	+/- NaN
	SIN, COS, TAN	IN = +/- INF или +/- NaN	+/- INF или +/- NaN
	ASIN, ACOS	IN за пределами допустимого диапазона от -1,0 до +1,0	+NaN
		IN = +/- NaN	+/- NaN
	ATAN	IN = +/- NaN	+/- NaN
	FRAC	IN = +/- INF или +/- NaN	+NaN
	EXPT	IN1 = +INF и IN2 не -INF	+INF
		IN1 отрицательный или -INF	+NaN, если IN2 = Real/LReal, в ином случае -INF
		IN1 или IN2 = +/- NaN	+NaN
		IN1 = 0,0 и IN2 с типом Real/LReal (только)	+NaN

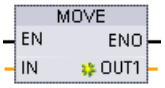
8.6 Инструкции для передачи данных

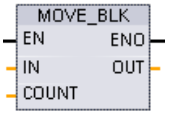
8.6.1 MOVE (копирование значения), MOVE_BLK (копирование области), UMOVE_BLK (непрерывное копирование области) и MOVE_BLK_VARIANT (копирование области)

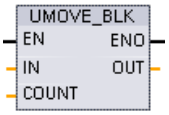
С помощью инструкции передачи выполняется копирование элементов данных в новую область памяти и конвертация из одного типа данных в другой. При этом исходные данные не меняются.

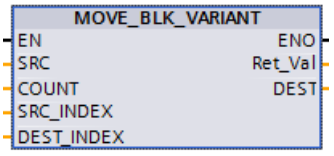
- С помощью инструкции MOVE копирует один элемент данных из адреса источника, заданного параметром IN, в целевой адрес, указанный с помощью параметра OUT.
- У инструкций MOVE_BLK и UMOVE_BLK дополнительно есть параметр COUNT. С помощью COUNT определяется, сколько элементов данных необходимо скопировать. Количество байт на скопированный элемент зависит от того, какой тип данных присвоен именам переменных (тегов) параметров IN и OUT в таблице переменных (тегов) PLC.

Таблица 8-68 Инструкции MOVE, MOVE_BLK, UMOVE_BLK иMOVE_BLK_VARIANT

LAD/FBD	SCL	Описание
	`out1 := in;`	Копирует элемент данных, хранящийся в назначенном адресе, в новый адрес или множество адресов.[1]

	`MOVE_BLK(`
	`in:=_variant_in,`
	`count:=_uint_in,`
	`out=>_variant_out);`
		Пересылка с возможностью прерывания, которая копирует область с элементами данных в новый адрес.

	`UMOVE_BLK(`
	`in:=_variant_in,`
	`count:=_uint_in,`
	`out=>_variant_out);`
		Пересылка без возможности прерывания, которая копирует область с элементами данных в новый адрес.

	`MOVE_BLK(`
	`SRC:=_variant_in,`
	`COUNT:=_udint_in,`
	`SRC_INDEX:=_dint_in,`
	`DEST_INDEX:=_dint_in,`
	`DEST=>_variant_out);`
		Копирует содержание исходной области памяти в целевую область памяти.
		Можно скопировать весь массив или элементы массива в другой массив с тем же типом данных. Размеры (количество элементов) исходного и целевого массива могут различаться. Можно скопировать несколько или один элемент массива. Используются типы данных Variant для указания на исходный и целевой массивы.

[1] Инструкция MOVE: Для добавления еще одного выхода в LAD или FBD кликнуть по символу "Создать" рядом с выходным параметром. В SCL используется несколько инструкций присвоения. Также можно использовать одну из цикловых конструкций.

Таблица 8-69 Типы данных для инструкции MOVE

Параметр	Тип данных	Описание
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Char, WChar, Array, Struct, DTL, Time, Date, TOD, IEC типы данных, типы данных PLC	Адрес источника
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Char, WChar, Array, Struct, DTL, Time, Date, TOD, IEC типы данных, типы данных PLC	Адрес получателя



Для добавления выходов MOVE кликнуть по символу "Создать" или правой кнопкой мыши по одному из параметров OUT на выходном соединителе и выбрать команду "Вставить выход".

Для удаления выхода кликнуть правой кнопкой мыши на входном соединении одного из параметров OUT (если имеется более двух исходных выходов) и выбрать команду "Удалить".

Таблица 8-70 Типы данных для инструкций MOVE_BLK и UMOVE_BLK

Параметр	Тип данных	Описание
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal Byte, Word, DWord, Time, Date, TOD, WChar	Начальный адрес источника
COUNT	UInt	Количество копируемых элементов данных
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, WChar	Начальный адрес получателя

Таблица 8-71 Типы данных для инструкции MOVE_BLK_VARIANT

Параметр	Тип данных	Описание
SRC	Variant (указывает на массив или отдельный элемент массива)	Исходная область для копирования
COUNT	UDInt	Количество копируемых элементов данных
SRC_INDEX	DInt	Индекс относительно нуля в массиве SRC
DEST_INDEX	DInt	Индекс относительно нуля в массиве DEST
RET_VAL	Int	Информация об ошибке
DEST	Variant (указывает на массив или отдельный элемент массива)	Целевая область, в который копируется содержимое исходной области.

Примечание

Правила для инструкций для передачи данных

- Для копирования тип данных Bool, использовать SET_BF, RESET_BF, R, S, или выходную катушку (LAD) (Страница 222).
- Для копирования данных одиночного элементарного типа данных следует использовать MOVE.
- Для копирования массива элементарного типа данных, следует использовать MOVE_BLK или UMOVE_BLK.
- Для копирования структуры, следует использовать MOVE.
- Для копирования строки, следует использовать S_MOVE (Страница 353).
- Для копирования отдельного символа в строке, следует использовать MOVE.
- Инструкции MOVE_BLK и UMOVE_BLK не могут быть использованы для копирования массивов или структур в I, Q, или M области памяти.

Инструкции MOVE_BLK и UMOVE_BLK отличаются методом обработки прерываний:

- Прерывающие события **помещаются в лист ожидания** и обрабатываются при выполнении MOVE_BLK. Инструкция MOVE_BLK используется тогда, когда данные на целевом адресе передачи не используются внутри подпрограммы OB прерывания или, в случае их использования, целевые данные могут быть не согласованными. Если операция MOVE_BLK прерывается, то последний переданный элемент данных на целевом адресе является полным и согласованным. Операция MOVE_BLK возобновляется после завершения выполнения OB прерывания.
- Прерывающие события **помещаются в лист ожидания, но обрабатываются** только по завершении выполнения UMOVE_BLK. Операция UMOVE_BLK используется тогда, когда операция передачи завершена и целевые данные должно быть согласованными до того, как подпрограмма выполнит OB прерывания. Для получения дополнительной информации см. Целостность данных (Страница 191).

ENO после выполнения операции MOVE всегда в состоянии ИСТИНА.

Таблица 8-72 Состояние ENO

ENO	**Условие**	**Результат**
1	Ошибки отсутствуют	Все элементы в COUNT были успешно скопированы.
0	Исходная область (IN) или целевая область (OUT) превышает доступную область памяти.	Подходящие по размеру элементы копируются. Неполные элементы не копируются.

Таблица 8-73 Коды условий для инструкции MOVE_BLK_VARIANT

RET_VAL (W#16#...)	**Описание**
0000	Ошибки отсутствуют
80B4	Несоответствие типов данных.
8151	Доступ к параметру SRC невозможен.
8152	Операнд параметра SRC имеет неверный тип данных.
8153	Ошибка при генерации кода в параметре SRC
8154	Операнд параметра SRC имеет тип данных Bool.
8281	Неверное значение параметра COUNT.
8382	Значение параметра SRC_INDEX лежит вне предельного значения для Variant.
8383	Значение параметра SRC_INDEX превышает верхнее предельное значение для массива.
8482	Значение параметра DEST_INDEX лежит вне предельного значения для Variant.
8483	Значение параметра DEST_INDEX превышает верхнее предельное значение для массива.
8534	Параметр DEST защищен от записи.
8551	Доступ к параметру DEST невозможен.
8552	Операнд параметра DEST имеет неверный тип данных.
8553	Ошибка при генерации кода в параметре DEST

RET_VAL (W#16#...)	Описание
8554	Операнд параметра DEST имеет тип данных Bool.
**Коды ошибок могут отображаться в редакторе тестов программ в виде целых или шестнадцатеричных чисел.	

8.6.2 Десериализация

С помощью инструкции "Десериализация" можно выполнить обратное преобразование последовательного представления типа данных PLC (UDT) в тип данных PLC и заполнить его содержимое. Если сравнение TRUE, то выход блока TRUE

Область памяти, в которой находится последовательное представление типа данных PLC, должна иметь тип данных "Array of Byte" и блок данных должен быть объявлен для стандартного доступа (не оптимизированный доступ). Убедиться, что имеется достаточное количество памяти до выполнения преобразования.

Инструкция позволяет преобразовать несколько последовательные представлений преобразованных типов данных PLC назад в их исходные типы данных.

Примечание

Если необходимо выполнить обратное преобразование только одного последовательного представления типа данных PLC (UDT), то можно использовать инструкцию "TRCV: Получить данные через коммуникационное соединение".

Таблица 8-74 Инструкция DESERIALIZE

LAD/FBD	SCL	Описание

	`ret_val := Deserialize(`
	`  SRC_ARRAY:=_variant_in_,`
	`  DEST_VARIABLE=>_variant_out_`
	`,`
	`  POS:=_dint_inout_);`
		Обратное преобразование последовательного представления типа данных PLC (UDT) в тип данных PLC и заполнение всего его содержимого.

Таблица 8-75 Параметры для инструкции DESERIALIZE

Параметр	Тип	Тип данных	Описание
SRC_ARRAY	IN	Variant	Глобальный блок данных с потоком данных
DEST_VARIABLE	INOUT	Variant	Переменная (тег), в который должен быть сохранен преобразованный тип данных PLC (UDT)

Параметр	Тип	Тип данных	Описание
POS	INOUT	DInt	Количество байтов, которые использует преобразованный тип данных PLC
RET_VAL	OUT	Int	Информация об ошибке

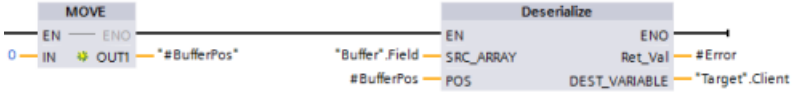
Таблица 8-76 Параметр RET_VAL

RET_VAL* (W#16#...)	Описание
0000	Ошибки отсутствуют
80B0	Перекрытие областей памяти для параметров SRC_ARRAY и DEST_VARIABLE.
8136	Блок данных в параметре DEST_VARIABLE не является блоком со стандартным доступом.
8150	Тип данных Variant параметра SRC_ARRAY не содержит значения.
8151	Ошибка при генерации кода в параметре SRC_ARRAY.
8153	Недостаточно свободной памяти для параметра SRC_ARRAY.
8250	Тип данных Variant параметра DEST_VARIABLE не содержит значения.
8251	Ошибка при генерации кода в параметре DEST_VARIABLE.
8254	Недействительный тип данных для параметра DEST_VARIABLE.
8382	Значение параметра POS лежит вне предельного значения для массива.
*Коды ошибок могут отображаться в редакторе тестов программ в виде целых или шестнадцатеричных чисел.	

Пример: Инструкция "Десериализация"

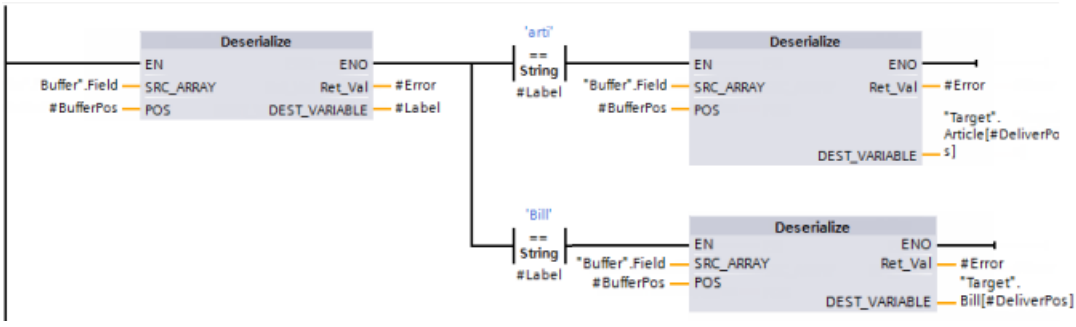
В следующем примере показан принцип работы данной инструкции:

Сегмент 1:



С помощью инструкции "MOVE" значение "0" копируется в переменную (тег) блока данных "#BufferPos". Инструкция Deserialize затем выполняет преобразование последовательного представления пользовательских данных из блока данных "Buffer" и записывает его в блок данных "Target" Инструкция Deserialize вычисляет количество байтов, которые используются преобразованными данными, и сохраняет его в переменной (теге) блока данных "#BufferPos".

Сегмент 2:



Инструкция "Deserialize" выполняет преобразование последовательного представления потока данных, на который указывает "Buffer" и записывает символы в операнд "#Label". Логика сравнивает символы, используя инструкции сравнения "arti" и "Bill". Если сравнение для "arti" = TRUE, данные являются данными об изделиях, которые будут преобразованы и записаны в структуру данных "Article" блока данных "Target". Если сравнение для "Bill" = TRUE, данные являются данными фактурирования, которые будут преобразованы и записаны в структуру данных "Bill" блока данных "Target".

Интерфейс функционального блока (или интерфейс функции):

Name	Data type
Input	
DeliverPos	Int
Output	
InOut	
Static	
Temp	
BufferPos	DInt
Error	Int
Label	String[4]

Пользовательские типы данных PLC:

Структуры двух типов данных PLC (UDT) для этого примера представлены ниже:

Article

	Name	Data type
1	Number	DInt
2	Declaration	String
3	Colli	Int

Client

	Name	Data type
1	Title	Int
2	Firstname	String[10]
3	Surname	String[10]

Блоки данных:

Оба блока данных для этого примера представлены ниже:

Target

	Name	Data type
1	Static	
2	Client	"Client"
3	Article	Array[0..10] of "Article"
4	Bill	Array[0..10] of Int

Buffer

	Name	Data type
1	Static	
2	Field	Array[0..294] of Byte

8.6.3 Serialize

С помощью инструкции "Serialize" можно преобразовать несколько типов данных PLC (UDT) в последовательное представление без каких-либо структурных потерь.

Инструкция может использоваться для временного помещения нескольких структурированных элементов данных из программы в буфер, например, в глобальный блок данных, чтобы после передать их в другой CPU. Область памяти, в которую сохраняются преобразованные типы данных PLC, должен иметь тип данных "ARRAY of BYTE" со стандартным типом доступа. Убедиться, что имеется достаточное количество памяти до выполнения преобразования.

Параметр POS содержит информацию о количестве байтов, которое используют преобразованные типы данных PLC.

Примечание

Для передачи одного единственного типа данных PLC (UDT) можно использовать инструкцию "TSEND: Получить данные через коммуникационное соединение".

Таблица 8-77 Инструкция SERIALIZE

LAD/FBD	SCL	Описание

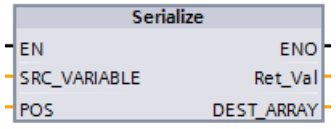
	`ret_val := Serialize(`
	`  SRC_VARIABLE=>_variant_in_,`
	`  DEST_ARRAY:=_variant_out_,`
	`  POS:=_dint_inout_);`
		Преобразует тип данных PLC (UDT) в последовательное представление.

Таблица 8-78 Параметры для инструкции SERIALIZE

Параметр	Тип	Тип данных	Описание
SRC_VARIABLE	IN	Variant	Тип данных PLC (UDT), который должен быть преобразован в последовательное представление
DEST_ARRAY	INOUT	Variant	Блок данных, в который должен быть сохранен сгенерированный поток данных
POS	INOUT	DInt	Количество байтов, которое используют преобразованные типы данных PLC. Параметр POS рассчитывается относительно нуля.
RET_VAL	OUT	Int	Информация об ошибке

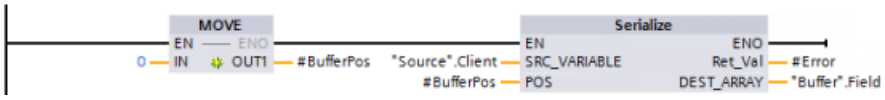
Таблица 8-79 Параметр RET_VAL

RET_VAL* (W#16#...)	Описание
0000	Ошибки отсутствуют
80B0	Перекрытие областей памяти для параметров SRC_VARIABLE и DEST_ARRAY.
8150	Тип данных Variant параметра SRC_VARIABLE не содержит значения.
8152	Ошибка при генерации кода в параметре SRC_VARIABLE.
8236	Блок данных в параметре DEST_ARRAY не является блоком со стандартным доступом.
8250	Тип данных Variant параметра DEST_ARRAY не содержит значения.
8252	Ошибка при генерации кода в параметре DEST_ARRAY.
8253	Недостаточно свободной памяти для параметра DEST_ARRAY.
8254	Недействительный тип данных для параметра DEST_VARIABLE.
8382	Значение параметра POS лежит вне предельного значения для массива.
*Коды ошибок могут отображаться в редакторе тестов программ в виде целых или шестнадцатеричных чисел.	

Пример: Инструкция Serialize

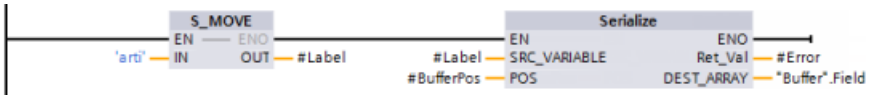
В следующем примере показан принцип работы данной инструкции:

Сегмент 1:



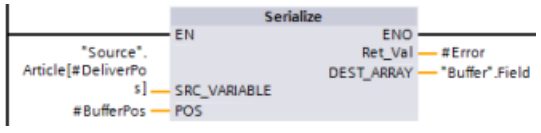
С помощью инструкции "MOVE" значение "0" копируется в параметр "#BufferPos". Инструкция "Serialize" выполняет последовательное преобразование пользовательских данных из блока данных "Source" в и записывает их в последовательном представлении в блок данных "Buffer" Инструкция сохраняет количество используемых последовательным представлением байт в параметр "#BufferPos".

Сегмент 2:



Далее логика вставляет разделительный текст, чтобы упростить обратное преобразование последовательного представления. Инструкция "S_MOVE" копирует строку "arti в параметр "#Label". Инструкция "Serialize" записывает эти символы после исходных данных в блок данных "Buffer". Инструкция добавляет число байтов в текстовой строке "arti" к уже сохраненному в параметре "#BufferPos" количеству.

Сегмент 3:



Инструкция "Serialize" преобразует данные определенного изделия, которые вычисляются во времени выполнения, из блока данных "Source" и записывает их в последовательном представлении в блок данных "Buffer" после символов "arti".

Интерфейс блока:

	Name	Data type
	Input	
	DeliverPos	Int
	Output	
	InOut	
	Static	
	Temp	
	BufferPos	DInt
	Error	Int
	Label	String[4]

Пользовательские типы данных PLC:

Структуры двух типов данных PLC (UDT) для этого примера представлены ниже:

Article

	Name	Data type
1	Number	DInt
2	Declaration	String
3	Colli	Int

Client

	Name	Data type
1	Title	Int
2	Firstname	String[10]
3	Surname	String[10]

Блоки данных:

Оба блока данных для этого примера представлены ниже:

Source

	Name	Data type
1	Static	
2	Client	"Client"
3	Article	Array[0..10] of "Article"

Buffer

	Name	Data type
1	Static	
2	Field	Array[0..294] of Byte

8.6.4 Инструкции FILL_BLK (заполнить блок) и UFILL_BLK (заполнить блок без прерываний)

Таблица 8-80 Инструкции FILL_BLK und UFILL_BLK

LAD/FBD	SCL	Описание

	`FILL_BLK(`
	`  in:=_variant_in,`
	`  count:=int,`
	`  out=>_variant_out);`
		Инструкция заполнения с возможностью прерывания: Заполняет диапазон адресов копиями назначенного элемента данных.

	`UFILL_BLK(`
	`  in:=_variant_in,`
	`  count:=int,`
	`  out=>_variant_out);`
		Инструкция заполнения без прерываний: Заполняет диапазон адресов копиями назначенного элемента данных.

Таблица 8-81 Типы данных для параметров

Параметр	Тип данных	Описание
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Адрес источника данных
COUNT	UDint, USInt, UInt	Количество копируемых элементов данных
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Адрес цели передачи данных

Примечание

Правила для операций заполнения данных

- Для копирования типа данных BOOL следует использовать SET_BF, RESET_BF, R, S или выходною катушку (LAD).
- Для заполнения одиночным элементарным типом данных, следует использовать MOVE.
- Для заполнения массива элементарным типом данных, следует использовать FILL_BLK или UFILL_BLK.
- Для копирования отдельного символа в строке, следует использовать MOVE.
- Инструкции FILL_BLK иUFILL_BLK не могут быть использованы для заполнения массивов в областях памяти I, Q, или M.

С помощью инструкций FILL_BLK и UFILL_BLK элемент исходных данных IN копируется в целевой адрес, при этом начальный адрес установлен в параметре OUT. Процесс копирования и заполнения соседних адресов продолжается до тех пор, пока количество копий не будет равно параметру COUNT.

Инструкции FILL_BLK и UFILL_BLK отличаются методом обработки прерываний:

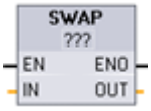
- Прерывающие события **помещаются в лист ожидания** и обрабатываются при выполнении FILL_BLK. Инструкция FILL_BLK используется тогда, когда данные на целевом адресе передачи не используются внутри подпрограммы OB прерывания или, в случае их использования, целевые данные могут быть не согласованными.
- Прерывающие события **помещаются в лист ожидания, но обрабатываются** только по завершении выполнения UFILL_BLK. Операция UFILL_BLK используется тогда, когда операция передачи завершена и целевые данные должно быть согласованными до того, как подпрограмма выполнит OB прерывания.

Таблица 8-82 Состояние ENO

ENO	Условие	Результат
1	Ошибки отсутствуют	Элемент IN был успешно скопирован во все цели COUNT.
0	Целевая область (OUT) превышает доступную область памяти	Подходящие по размеру элементы копируются. Неполные элементы не копируются.

8.6.5 SWAP (поменять месторасположение)

Таблица 8-83 Инструкция SWAP

LAD/FBD	SCL	Описание
	`out := SWAP(in);`	Меняет порядок байтов для двух или четырех байтовых элементов данных. При этом не происходят изменения в порядке битов самого байта. ENO после выполнения инструкции SWAP всегда имеет состояние TRUE.

[1] В LAD и FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.

Таблица 8-84 Типы данных для параметров

Параметр	Тип данных	Описание
IN	Word, DWord	Нормально расположенные байты данных в параметре IN
OUT	Word, DWord	Измененный порядок расположения байтов данных в параметре OUT

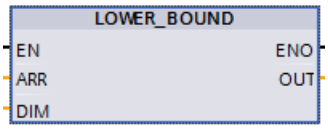
Пример 1	Параметр IN = MB0 (до выполнения)		Параметр OUT = MB4, (после выполнения)	
Адрес	MW0	MB1	MW4	MB5
W#16#1234	12	34	34	12
WORD	MSB	LSB	MSB	LSB

Пример 2	Параметр IN = MB0 (до выполнения)				Параметр OUT = MB4, (после выполнения)			
Адрес	MD0	MB1	MB2	MB3	MD4	MB5	MB6	MB7
DW#16# 12345678	12	34	56	78	78	56	34	12
DWORD	MSB			LSB	MSB			LSB

8.6.6 LOWER_BOUND: (считывание нижней границы массива ARRAY)

Таблица 8-85 Инструкция LOWER_BOUND

LAD/FBD	SCL	Описание

	`out := LOWER_BOUND(`
	`ARR:=_variant_in_,`
	`DIM:=_udint_in_);`
		Возможность объявления переменных (тегов) с ARRAY[*] на интерфейсе блока. Для этих локальных переменных (тегов) возможно считывание предельных значений массива ARRAY. Необходимо указать требуемую размерность в параметре DIM.
		С помощью инструкции LOWER_BOUND (считывание нижней границы массива ARRAY) возможно считывание нижнего предельного значения переменной (тега) массива.

Параметры

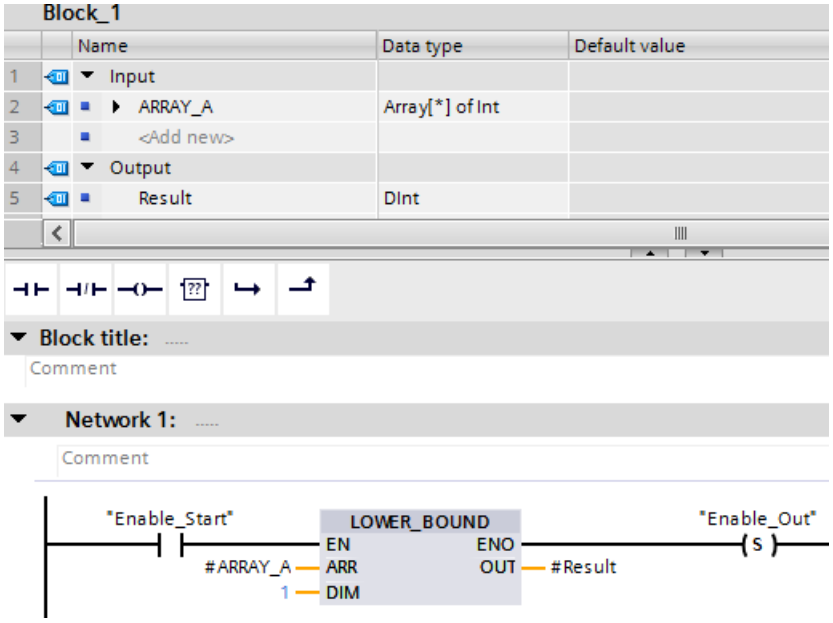
В приведенной ниже таблице показаны параметры инструкции LOWER_BOUND: Считывание нижней границы массива ARRAY

Параметры	Объявление	Тип данных	Область памяти	Описание
EN	Input	BOOL	I, Q, M, D, L	Разрешающий вход
ENO	Output	BOOL	I, Q, M, D, L	Разрешающий выход ENO имеет состояние сигнала "0", если имеет место одно из следующих условий: • Разрешающий входEN имеет состояние сигнала "0". • Указанная на входе DIM размерность отсутствует.
ARR	Input	ARRAY [*]	FB: Область InOut FC: Области Input и InOut	Массив ARRAY, нижняя граница переменных которого должна быть считана.
DIM	Input	UDINT	I, Q, M, D, L или постоянная	Размерность массива ARRAY, нижняя граница переменных которого должна быть считана.
OUT	Output	DINT	I, Q, M, D, L	Результат

Дополнительную информацию о допустимых типах данных можно найти в разделе "Типы данных (Страница 123)":

Пример

На интерфейсе блока функции (FC) входной параметр ARRAY_A представляет собой одномерный массив с переменными размерами.



Если операнд "Enable_Start" выводит значение сигнала "1", то CPU выполняет инструкцию LOWER_BOUND . CPU считывает переменный нижний предел массива #ARRAY_A из одномерного массива. Если инструкция выполняется с ошибками, то устанавливается операнд "Enable_Out"и операнд "Result" устанавливается на нижний предел массива.

8.6.7 UPPER_BOUND: (считывание верхней границы массива ARRAY)

Таблица 8-86 Инструкция LOWER_BOUND

LAD/FBD	SCL	Описание

UPPER_BOUND
EN ENO
ARR OUT
DIM
	`out := UPPER_BOUND(`
	`ARR:=_variant_in_,`
	`DIM:=_udint_in_);`
		Возможность объявления переменных (тегов) с ARRAY[*] на интерфейсе блока. Для этих локальных переменных (тегов) возможно считывание предельных значений массива ARRAY. Необходимо указать требуемую размерность в параметре DIM.
		С помощью инструкции UPPER_BOUND (считывание верхней границы массива ARRAY) возможно считывание нижнего предельного значения переменной (тега) массива ARRAY.

Параметры

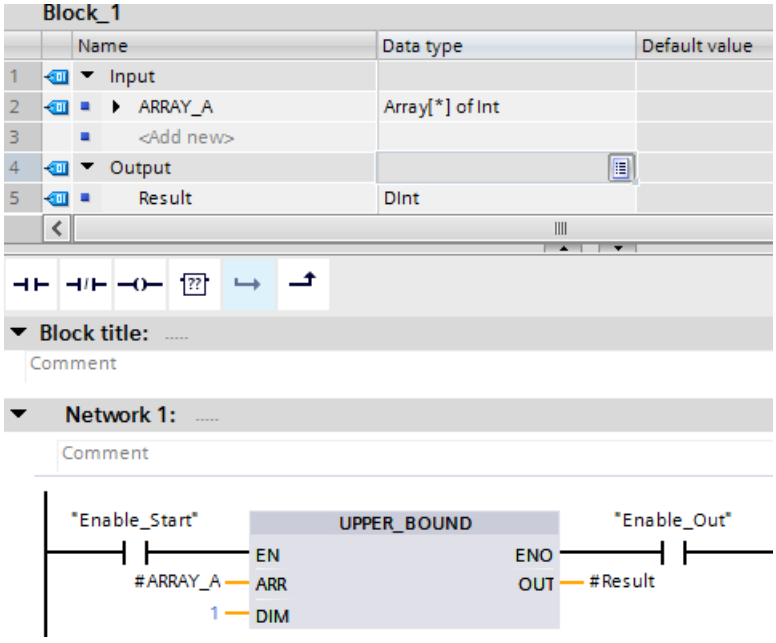
В приведенной ниже таблице показаны параметры инструкции UPPER_BOUND: (считывание верхней границы массива ARRAY):

Параметры	Объявление	Тип данных	Область памяти	Описание
EN	Input	BOOL	I, Q, M, D, L	Разрешающий вход
ENO	Output	BOOL	I, Q, M, D, L	Разрешающий выход
ARR	Input	ARRAY [*]	FB: Область InOut FC: Области Input и InOut	Массив ARRAY, верхняя граница переменных которого должна быть считана.
DIM	Input	UDINT	I, Q, M, D, L или постоянная	Размерность массива ARRAY, верхняя граница переменных которого должна быть считана.
OUT	Output	DINT	I, Q, M, D, L	Результат

Дополнительную информацию о допустимых типах данных можно найти в разделе "Типы данных (Страница 123)":

Пример

На интерфейсе блока функции (FC) входной параметр ARRAY_A представляет собой одномерный массив с переменными размерами.



Если операнд "Enable_Start" выводит значение сигнала "1", то CPU выполняет инструкцию. CPU считывает переменный верхний предел массива #ARRAY_A из одномерного массива. Если инструкция выполняется с ошибками, то устанавливаются операнд "Enable_Out" и операнд "Result".

8.6.8 Инструкции чтения / записи памяти

8.6.8.1 PEEK и POKE (только SCL)

SCL предлагает инструкции PEEK и POKE, которые позволяют выполнять чтение/запись из/в блоки данных, ввод-вывод или память. Для операции указываются параметры для определенных байтовых или битовых смещений.

Примечание

Для использования инструкций PEEK и POKE с блоками данных потребуются стандартные (не оптимизированные) блоки данных. Также обратите внимание на то, что инструкции PEEK и POKE просто передают данные. У них нет информации о типах данных по указанным адресам.

```
PEEK(area:=_in_,
     dbNumber:=_in_,
     byteOffset:=_in_);
```

Читает байт, на который ссылается byteOffset, из указанного блока данных, из I/O или области памяти.

Пример ссылки на блок данных:

```
%MB100 := PEEK(area:=16#84,
dbNumber:=1, byteOffset:=#i);
```

Пример ссылки на вход IB3:

```
%MB100 := PEEK(area:=16#81,
dbNumber:=0, byteOffset:=#i); // when
#i = 3
```

```
PEEK_WORD(area:=_in_,
     dbNumber:=_in_,
     byteOffset:=_in_);
```

Читает слово, на которое ссылается byteOffset, из указанного блока данных, из I/O или области памяти.

Пример:

```
%MW200 := PEEK_WORD(area:=16#84,
dbNumber:=1, byteOffset:=#i);
```

```
PEEK_DWORD(area:=_in_,
     dbNumber:=_in_,
     byteOffset:=_in_);
```

Читает двойное слово, на которое ссылается byteOffset, из указанного блока данных, из I/O или области памяти.

Пример:

```
%MD300 := PEEK_DWORD(area:=16#84,
dbNumber:=1, byteOffset:=#i);
```

```
PEEK_BOOL(area:=_in_,
     dbNumber:=_in_,
     byteOffset:=_in_,
     bitOffset:=_in_);
```

Читает булево значение слово, на которое ссылается bitOffset и byteOffset, из указанного блока данных, из I/O или области памяти.

Пример:

```
%MB100.0 := PEEK_BOOL(area:=16#84,
dbNumber:=1, byteOffset:=#ii,
bitOffset:=#j);
```

```
POKE(area:=_in_,
     dbNumber:=_in_,
     byteOffset:=_in_,
     value:=_in_);
```

Записывает значение (байт, слово или двойное слово) в указанный byteOffset указанного блока данных, I/O или области памяти.

Пример ссылки на блок данных:

```
POKE(area:=16#84, dbNumber:=2,
byteOffset:=3, value:="Tag_1");
```

Пример ссылки на выход QB3:

```
POKE(area:=16#82, dbNumber:=0,
byteOffset:=3, value:="Tag_1");
```

```
POKE_BOOL(area:=_in_,
     dbNumber:=_in_,
     byteOffset:=_in_,
     bitOffset:=_in_,
     value:=_in_);
```

Записывает булево значение в указанный bitOffset и byteOffset указанного блока данных, I/O или области памяти.

Пример:

```
POKE_BOOL(area:=16#84, dbNumber:=2,
byteOffset:=3, bitOffset:=5,
value:=0);
```

```
POKE_BLK(area_src:=_in_,
     dbNumber_src:=_in_,
     byteOffset_src:=_in_,
     area_dest:=_in_,
     dbNumber_dest:=_in_,
     byteOffset_dest:=_in_,
     count:=_in_);
```

Записывает заданное в "count" число байтов, начиная с указанного byteOffset указанного исходного блока данных, исходного I/O или исходной области памяти в указанный byteOffset целевого блока данных, целевого I/O или целевой области памяти.

Пример:

```
POKE_BLK(area_src:=16#84,
dbNumber_src:=#src_db,
byteOffset_src:=#src_byte,
area_dest:=16#84,
dbNumber_dest:=#src_db,
byteOffset_dest:=#src_byte,
count:=10);
```

Для инструкций PEEK и POKE действуют следующие значения для параметров "area", "area_src" и "area_dest". Для областей, отличных от блоков данных, параметр dbNumber должен быть всегда равен 0.

16#81	I
16#82	Q
16#83	M
16#84	DB

8.6.8.2 Инструкции чтения и записи формата Big- и Little-Endian (SCL)

S7-1200 CPU предлагает инструкции SCL для чтения и записи данных в формате Little-Endian и Big-Endian. Формат Little-Endian означает, что байт с самым младшим битом будет сохранен в самый младший адрес памяти. Формат Big-Endian означает, что байт с самым старшим битом будет сохранен в самый младший адрес памяти.

Четыре инструкции SCL для чтения и записи данных в формате Little-Endian и Big-Endian описаны ниже:

- READ_LITTLE (чтение данных в формате Little-Endian)
- WRITE_LITTLE (запись данных в формате Little-Endian)
- READ_BIG (чтение данных в формате Big-Endian)
- WRITE_BIG (запись данных в формате Big-Endian)

Таблица 8-87 Инструкции чтения и записи формата Big- и Little-Endian

LAD/FBD	SCL	Описание

Недоступно	`READ_LITTLE(`
	`  src_array:=_variant_in_,`
	`  dest_Variable =>_out_,`
	`  pos:=_dint_inout)`
		Считывает данные из области памяти и записывает их в одиночную переменную (тег) в байтовом формате Little-Endian.

Недоступно	`WRITE_LITTLE(`
	`  src_variable:=_in_,`
	`  dest_array =>_variant_inout_,`
	`  pos:=_dint_inout)`
		Записывает данные из одиночной переменной (тега) в область памяти в байтовом формате Little-Endian.

Недоступно	`READ_BIG(`
	`  src_array:=_variant_in_,`
	`  dest_Variable =>_out_,`
	`  pos:=_dint_inout)`
		Считывает данные из области памяти и записывает их в одиночную переменную (тег) в байтовом формате Big-Endian.

Недоступно	`WRITE_BIG(`
	`  src_variable:=_in_,`
	`  dest_array =>_variant_inout_,`
	`  pos:=_dint_inout)`
		Записывает данные из одиночной переменной (тега) в область памяти в байтовом формате Big-Endian.

Таблица 8-88 Параметры для инструкции READ_LITTLE and READ_BIG

Параметры	Тип данных	Описание
src_array	Байтовый массив	Область памяти из которой производится чтение данных
dest_Variable	Битовые строки, целые числа, вещественные числа, таймеры, дата и время, символьные строки	Целевая переменная для записи данных
pos	DINT	Позиция относительно нуля, с которой должно начинаться чтение данных со входа src_array.

Таблица 8-89 Параметры для инструкции WRITE_LITTLE and WRITE_BIG

Параметры	Тип данных	Описание
src_variable	Битовые строки, целые числа, числа с плавающей запятой, LDT, TOD, LTOD, DATA, Char, WChar	Исходные данные из переменной (тега)
dest_array	Байтовый массив	Область памяти, в которую будут записываться данные
pos	DINT	Позиция относительно нуля, с которой должна начинаться запись данных на выход dest_array.

Таблица 8-90 Параметр RET_VAL

RET_VAL* (W#16#...)	Описание
0000	Ошибки отсутствуют
80B4	SRC_ARRAY илиDEST_ARRAY это не Array of Byte
8382	Значение параметра POS лежит вне предельного значения для массива.
8383	Значение параметра POS в пределах допустимых значений массива, но размер области памяти превышает верхнюю границу массива.
*Коды ошибок могут отображаться в редакторе тестов программ в виде целых или шестнадцатеричных чисел.	

8.6.9 Инструкции Variant

8.6.9.1 VariantGet (чтение значения переменной (тега) VARIANT)

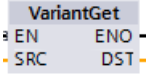
С помощью инструкции "Чтение значения переменной (тега) Variant" может быть считано значение переменной (тега), на которую ссылается указатель Variant в параметре SRC, и после записано в переменную (тег) параметра DST.

Параметр SRC имеет тип данных Variant. Для параметра DST может быть запрограммирован любой тип данных, кроме Variant.

Тип данных переменной (тега) в параметре DST должен совпадать с типом данных, на который указывает Variant.

Таблица 8-91 Инструкция VariantGet

LAD/FBD	SCL	Описание

	`VariantGet(`
	`  SRC:=_variant_in_,`
	`  DST=>_variant_out_);`
		Считывает переменную (тег), на которую указывает параметр SRC, и записывает ее в переменную (тег) в параметре DST.

Примечание

Для копирования структур и массивов можно использовать инструкцию "MOVE_BLK_VARIANT: Копирование области"

Таблица 8-92 Параметры для инструкции VariantGet

Параметры	Тип данных	Описание
SRC	Variant	Указатель на исходные данные
DST	Битовые строки, целые числа, числа с плавающей точкой, таймеры, дата и время, символьные строки, элементы массивов, типы данных PLC	Цель для записи данных

Таблица 8-93 Состояние ENO

ENO	Условие	Результат
1	Ошибки отсутствуют	Инструкция скопировала данные переменной (тега), на которую указывает SRC, в переменную DST.
0	Разрешающий вход EN имеет состояние сигнала "0" или несоответствие типов данных.	Инструкция не скопировала данных.

8.6.9.2 VariantPut (запись значения переменной (тега) VARIANT)

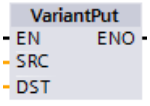
С помощью инструкции "Запись значения переменной (тега) VARIANT" можно записать значение переменной (тега) в параметре SRC в переменную (тег) в параметре DST, на который указывает VARIANT.

Параметр DST имеет тип данных VARIANT. Для параметра SRC может быть запрограммирован любой тип данных, кроме VARIANT.

Тип данных переменной (тега) в параметре SRC должен совпадать с типом данных, на который указывает VARIANT.

Таблица 8-94 Инструкция VariantPut

LAD/FBD	SCL	Описание

	`VariantPut(`
	`  SRC:=_variant_in_,`
	`  DST=>_variant_in_);`
		Записывает адресованную с помощью параметра SRC переменную (тег) в элемент типа Variant, на который указывает параметр DST.

Примечание

Для копирования структур и массивов можно использовать инструкцию "MOVE_BLK_VARIANT: Копирование области".

Таблица 8-95 Параметры для инструкции VariantPut

Параметры	Тип данных	Описание
SRC	Bit strings, integers, floating-point numbers, timers, date and time, character strings, ARRAY elements, PLC data types	Указатель на исходные данные
DST	Variant	Цель для записи данных

Таблица 8-96 Состояние ENO

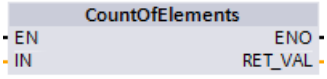
ENO	Условие	Результат
1	Ошибки отсутствуют	Инструкция скопировала данные перемен-ной (тега) SRC в переменную (тег)DST.
0	Разрешающий вход EN имеет состояние сигнала "0" или несоответствие типов данных.	Инструкция не скопировала данных.

8.6.9.3 CountOfElements (запрос количества элементов массива ARRAY)

С помощью инструкции "Запрос количества элементов массива ARRAY" можно узнать, сколько элементов Array имеется в переменной (теге), на которую указывает элемент Variant.

Если массив ARRAY одномерный, то инструкция вернет разницу между верхним и нижним предельным значением +1. При работе с многомерным массивом ARRAY, инструкция вернет произведение всех измерений.

Таблица 8-97 Инструкция CountOfElements

LAD/FBD	SCL	Описание
	`Result := CountOfElements( _variant_in_);`	Подсчитывает количество элементов в массиве, на который указывает параметр IN.

Примечание

Если Variant указывает на Array of Bool, инструкция подсчитывает элементы заполнения до ближайшей границы байта. Для Array[0..1] of Bool, к примеру, возвращается значение 8.

Таблица 8-98 Параметры для инструкции CountOfElements

Параметры	Тип данных	Описание
IN	Variant	Переменная (тег) с элементами массива для подсчета
RET_VAL	UDint	Результат инструкции

Таблица 8-99 Состояние ENO

ENO	Условие	Результат
1	Ошибки отсутствуют	Инструкция возвращает количество элементов массива.
0	Разрешающий вход EN имеет состояние сигнала "0" или Variant указывает не на массив.	Инструкция возвращает 0.

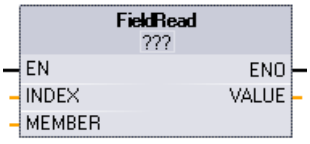
8.6.10 Унаследованные инструкции

8.6.10.1 Инструкции FieldRead (чтение поля) и FieldWrite (запись поля)

Примечание

STEP 7 V10.5 **не поддерживает** ссылку на переменную в качестве индекса массива или многомерные массивы. Инструкции FieldRead и FieldWrite использовались, чтобы обеспечить операции с переменным индексом для одномерного массива. STEP 7 от версии 11 и выше **поддерживает** переменную в качестве индекса массива и многомерные массивы. FieldRead и FieldWrite включены в STEP 7 версии 11 и выше для совместимости с программами предыдущих версий, которые использовали эти инструкции.

Таблица 8-100 Инструкции FieldRead и FieldWrite

LAD/FBD	SCL	Описание
	`value := member[index];`	FieldRead считывает элемент массива со значением индекса INDEX из массива, первый элемент которого задан в параметре MEMBER. Значение элемента массива далее передается по указанному в параметре VALUE адресу.
	`member[index] := value;`	WriteField передает значение по указанному в параметре VALUE адресу в массив, первый элемент которого указывается параметром MEMBER. Значение передается в элемент массива с индексом, задаваемым параметром INDEX.

[1] В LAD и FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.

Таблица 8-101 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
Индекс	Вход	DInt	Индекс элемента массива, который необходимо прочитать или в который необходимо выполнить запись.
Member [1]	Вход	Двоичные числа, целые числа, числа с плавающей запятой, таймеры, DATE, TOD, CHAR и WCHAR как компоненты переменной (тега) ARRAY	Адрес первого элемента в одномерном массиве, определенном в глобальном блоке данных или на интерфейсе блока. Пример: Если индекс массива указывается как [-2..4], то индекс первого элемента равен -2, а не 0.
Value [1]	Выход	Двоичные числа, целые числа, числа с плавающей запятой, таймеры, DATE, TOD, CHAR, WCHAR	Адрес, куда будет скопирован указанный элемент массива (FieldRead). Адрес значения, которое будет скопировано в указанный элемент массива (FieldWrite).

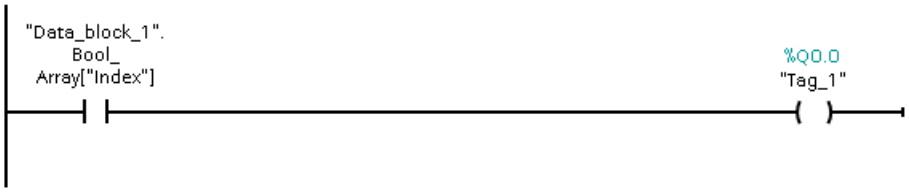
[1] Типы данных указанного параметром MEMBER элемента массива и параметра VALUE должны совпадать.

Разрешающий выход равен "0", если имеет место одно из следующих условий:

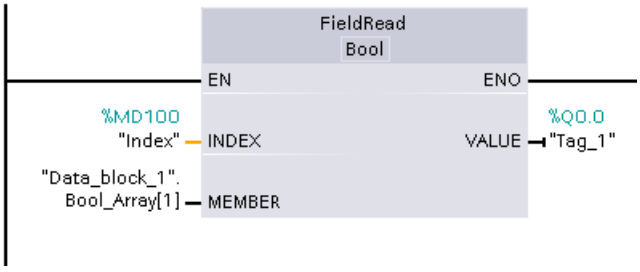
- Вход EN имеет состояние сигнала "0".
- Элемент массива, указанный в параметре INDEX, не определен в массиве, на который указывает параметр MEMBER.
- В процессе обработки возникают ошибки, например переполнение.

Пример: Обращение к данным посредством индексации массива

Для получения доступа к элементам массива через переменную, достаточно использовать переменную в качестве индекса массива в логике программы. Например, в сегменте ниже выход устанавливается согласно булевому значению из массива в "Data_block_1", определяемому посредством переменной (тега) PLC "Index".



Логика с индексом массива из переменных соответствует прежнему методу с использованием инструкции FieldRead:



Инструкции FieldWrite и FieldRead могут быть заменены логикой индексации массива через переменные.

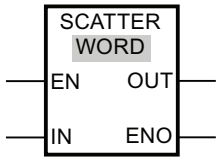
В SCL отсутствуют инструкции FieldRead или FieldWrite, но поддерживается косвенная адресация массива с помощью переменной:
`#Tag_1 := "Data_block_1".Bool_Array[#Index];`

8.6.11 SCATTER

SCATTER: Разбивка битовой последовательности на отдельные биты

Инструкция для разбивки битовой последовательности на отдельные биты делит переменную (тег) типа данных BYTE, WORD или DWORD на отдельные биты и сохраняет их в тип данных ARRAY из BOOL, тип данных "anonyme STRUCT" или тип данных PLC исключительно с булевыми элементами.

Таблица 8-102 SCATTER

LAD/FBD	SCL	Описание
	**`SCATTER (IN := #SourceWord, OUT => #DestinationArray );`**	SCATTER: Инструкция для разбивки битовой последовательности на отдельные биты делит переменную (тег) типа данных BYTE, WORD или DWORD на отдельные биты и сохраняет их в тип данных ARRAY из BOOL, тип данных "anonyme STRUCT" или тип данных PLC исключительно с булевыми элементами.

Примечание

Многомерный массив ARRAY из BOOL

В инструкции для разбивки битовой последовательности на отдельные биты нельзя использовать многомерный массив ARRAY из BOOL.

Примечание

Длина типа данных ARRAY, STRUCT или PLC

Число элементов в типе данных ARRAY, anonyme STRUCT или PLC должно точно соответствовать указанному в битовой последовательности числу элементов. Для типа данных BYTE это означает, к примеру, что тип данных ARRAY, STRUCT или PLC должен иметь точно 8 элементов (WORD = 16 и DWORD = 32).

Примечание

Примечание о доступности инструкции

Инструкция может использоваться с CPU серии S7-1200 от версии прошивки 4.2 и с CPU серии S7-1500 с версией прошивки 2.1.

Можно, например, выполнить анализ слова состояния и прочесть и изменить статус отдельных битов с помощью индекса. С помощью GATHER биты снова могут быть собраны в битовую последовательность.

Разрешающий выход ENO возвращает состояние сигнала "0", если имеет место одно из следующих условий:

- Разрешающий вход EN имеет состояние сигнала "0".
- Тип данных ARRAY, STRUCT или PLC не предоставляет достаточного количества элементов BOOL.

Типы данных для инструкции SCATTER

В приведенной ниже таблице показаны параметры инструкции:

Параметры	Объявление	Тип данных		Область памяти	Описание
		S7-1200	S7-1500		
EN	Input	BOOL	BOOL	I, Q, M, D, L или постоянная	Разрешающий вход
ENO	Output	BOOL	BOOL	I, Q, M, D, L	Разрешающий выход
IN	Input	BYTE, WORD, DWORD	BYTE, WORD, DWORD	I, Q, M, D, L	Битовая последовательность для анализа. Значения не могут находиться в области ввода-вывода или в DB технологического объекта.
OUT	Output	Тип данных ARRAY[*] из BOOL, STRUCT или PLC *: 8, 16, 32 или 64 элемента	Тип данных ARRAY[*] из BOOL, STRUCT или PLC *: 8, 16, 32 или 64 элемента	I, Q, M, D, L	Тип данных ARRAY, STRUCT или PLC, в который сохраняются отдельные биты

Дополнительную информацию о действующих типах данных можно найти в разделе "См. также".

Пример с массивом ARRAY

Создать следующие переменные (теги) на интерфейсе блока:

Переменная (тег)	Область	Тип данных
Enable	Вход	BOOL
SourceWord		WORD
EnableOut	Выход	BOOL
DestinationArray		ARRAY[0..15] из BOOL

В следующем примере показан принцип работы данной инструкции:

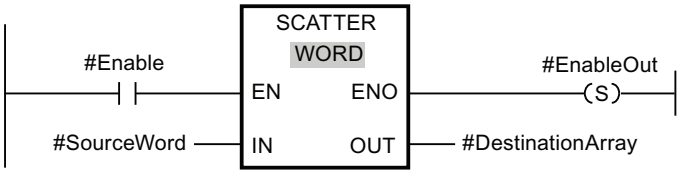


Рисунок 8-1 SCATTER - Пример 2

В нижеследующей таблице показано, как инструкция работает со специальными значениями операндов:

Параметры	Операнд	Тип данных
IN	SourceWord	WORD (16 битов)
OUT	DestinationUDT	Операнд "DestinationUDT" имеет тип данных PLC (UDT). Он состоит из 16 элементов и тем самым имеет точно такой же размер, что и анализируемый тип данных WORD.

Если операнд " #Enable" возвращает на разрешающем входе EN состояние сигнала "1", то инструкция выполняется. Операнд #SourceWord типа данных WORD разбивается на отдельные биты (16) и назначается отдельным элементам операнда #DestinationArray. Если при выполнении инструкции возникает ошибка, то операнд #EnableOut выводит на разрешающий выход ENO состояние сигнала "0".

Дополнительную информацию и программный код для приведенного выше примера можно найти здесь: Библиотека образцов для инструкций.

Пример с типом данных PLC (UDT)

Создать следующий блок данных PLC "myBits":

myBits

	Name	Data type
1	GeneralError	Bool
2	DeviceError	Bool
3	CommError	Bool
4	myError1	Bool
5	myError2	Bool
6	myError3	Bool
7	myError4	Bool
8	myError5	Bool
9	myError6	Bool
10	myError7	Bool
11	myError8	Bool
12	myError9	Bool
13	myError10	Bool
14	myError11	Bool
15	myError12	Bool
16	myError13	Bool

Создать следующие переменные (теги) на интерфейсе блока:

Переменная (тег)	Область	Тип данных
Enable	Вход	BOOL
SourceWord		WORD
EnableOut	Выход	BOOL
DestinationUDT		"myBits"

В следующем примере показан принцип работы данной инструкции:

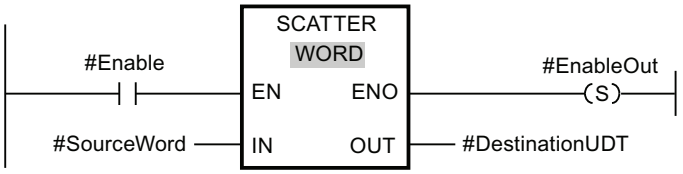


Рисунок 8-2 SCATTER - Пример

В нижеследующей таблице показано, как инструкция работает со специальными значениями операндов:

Параметр	Операнд	Тип данных
IN	SourceWord	WORD (16 битов)
OUT	DestinationUDT	Операнд "DestinationUDT" имеет тип данных PLC (UDT). Он состоит из 16 элементов и тем самым имеет точно такой же размер, что и анализируемый тип данных WORD.

Если операнд " #Enable" возвращает на разрешающем входе EN состояние сигнала "1", то инструкция выполняется. Операнд #SourceWord типа данных WORD разбивается на отдельные биты (16) и назначается отдельным элементам операнда #DestinationArray. Если при выполнении инструкции возникает ошибка, то операнд #EnableOut выводит на разрешающий выход ENO состояние сигнала "0".

См. также

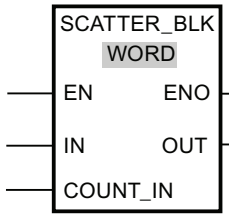
Новые функции (Страница 35)

8.6.12 SCATTER_BLK

SCATTER_BLK: Разбивка элементов массива ARRAY из битовых последовательностей на отдельные биты

Инструкция Разбивка элементов массива ARRAY из битовой последовательности на отдельные биты делит один или несколько элементов массива ARRAY из данных BYTE, WORD или DWORD на отдельные биты и сохраняет их в тип данных ARRAY из BOOL, тип данных "anonyme STRUCT" или тип данных PLC исключительно с булевыми элементами. В параметре COUNT_IN указывается, сколько элементов исходного массива ARRAY должно быть проанализировано. Исходный массив ARRAY в параметре IN может иметь больше элементов, чем указано в параметре COUNT_IN. Тип данных ARRAY из BOOL, anonyme STRUCT или PLC должен иметь достаточно элементов для сохранения битов анализируемых битовых последовательностей. Но и целевая область памяти может иметь больший размер.

Таблица 8-103 SCATTER_BLK

LAD/FBD	SCL	Описание
	**SCATTER_BLK(IN:=_byte_in_,** **COUNT_IN:=_uint_in_,** **OUT=>_bool_out_);_IN:=_uint_in_,**	Инструкция Разбивка элементов массива ARRAY из битовой последовательности на отдельные биты делит один или несколько элементов массива ARRAY из данных BYTE, WORD или DWORD на отдельные биты и сохраняет их в тип данных ARRAY из BOOL, тип данных "anonyme STRUCT" или тип данных PLC исключительно с булевыми элементами. В параметре COUNT_IN указывается, сколько элементов исходного массива ARRAY должно быть проанализировано. Исходный массив ARRAY в параметре IN может иметь больше элементов, чем указано в параметре COUNT_IN. Тип данных ARRAY из BOOL, anonyme STRUCT или PLC должен иметь достаточно элементов для сохранения битов анализируемых битовых последовательностей. Но и целевая область памяти может иметь больший размер.

Примечание

Данные НЕ записываются, если ENO = FALSE

У S7-1200 CPU только для инструкции SCATTER_BLK действует правило, что данных не записываются на выход, если ENO = FALSE.

Примечание

Многомерный массив ARRAY из BOOL

Если в случае ARRAY речь идет о многомерном массиве из BOOL, то при подсчете учитываются и биты заполнения размерностей, даже если они не были явно объявлены.

Пример 1: Массив ARRAY[1..10,0..4,1..2] из BOOL обрабатывается как ARRAY[1..10,0..4,1..8] из BOOL или как ARRAY[0..399] из BOOL.

Пример 2: На параметре IN подключается ARRAY[0..5] из WORD (sourceArrayWord[2]). Параметр COUNT_IN имеет значение "3". На параметре OUT подключается ARRAY[0..1,0..5,0..7] из BOOL (destinationArrayBool[0,0,0]). Как массив на параметре IN, так и массив на параметре OUT, имеют размер в 96 бит. Массив ARRAY из WORD разбивается на 48 отдельных битов.

Примечание

Если нижняя граница массива ARRAY целевого массива ARRAY не "0", то необходимо учитывать следующее:

Для поддержания производительности индекс всегда должен начинаться на границе BYTE, WORD или DWORD. Это значит, что расчет индекса должен начинаться на нижней границе массива ARRAY. За основу для этого расчета берется следующая формула:

Правильные индексы = Нижняя граница массива ARRAY + n(число битовых последовательностей) × число битов в нужной битовой последовательности

Для массива ARRAY[-2..45] из BOOL и битовой последовательности WORD расчет выглядит следующим образом:

- Правильный индекс (-2) = -2 + 0 × 16
- Правильный индекс (14) = -2 + 1 × 16
- Правильный индекс (30) = -2 + 2 × 16

Пример представлен ниже.

Примечание

Доступность инструкции

Инструкция может использоваться с CPU серии S7-1200 от версии прошивки 4.2 и с CPU серии S7-1500 с версией прошивки 2.1.

Можно, например, выполнить анализ слов состояния и прочесть и изменить статус отдельных битов с помощью индекса. С помощью GATHER биты снова могут быть собраны в битовую последовательность.

Разрешающий выход ENO возвращает состояние сигнала "0", если имеет место одно из следующих условий:

- Разрешающий вход EN имеет состояние сигнала "0".
- Исходный массив ARRAY имеет меньше элементов, чем указано в параметре COUNT_IN.

- Индекс целевого массива ARRAY не начинается на границе BYTE, WORD или DWORD. В этом случае результат не записывается в массив ARRAY из BOOL.
- Тип данных ARRAY[*] из BOOL, STRUCT или PLC не располагает необходимым числом элементов.
  - S7-1500 CPU: В этом случае выполняется анализ максимально возможного числа битовых последовательностей с записью в тип данных ARRAY из BOOL, anonyme STRUCT или PLC. Прочие битовые последовательности в дальнейшем не учитываются.
  - S7-1200 CPU: Копирование не выполняется.

Типы данных для инструкции SCATTER_BLK

В приведенной ниже таблице показаны параметры инструкции:

Параметр	Объявление	Тип данных		Область памяти	Описание
		S7-1200	S7-1500		
EN	Input	BOOL	BOOL	I, Q, M, D, L или постоянная	Разрешающий вход
ENO	Output	BOOL	BOOL	I, Q, M, D, L	Разрешающий выход
IN	Input	Элемент массива ARRAY[*] из <битовой последовательности>	Элемент массива ARRAY[*] из <битовой последовательности>	I, Q, M, D, L	Массив ARRAY из <битовой последовательности> для анализа. Значения не могут находиться в области ввода-вывода или в DB технологического объекта.

Параметр	Объявление	Тип данных		Область памяти	Описание
		S7-1200	S7-1500		
COUNT_IN	Input	USINT, UINT, UDINT	USINT, UINT, UDINT	I, Q, M, D, L	Счетчик для числа элементов исходного массива ARRAY для анализа. Значение не может находиться в области ввода-вывода или в базе данных технологического объекта.
OUT	Output	Элемент типа данных ARRAY[*] из BOOL, STRUCT или PLC	Элемент типа данных ARRAY[*] из BOOL, STRUCT или PLC	I, Q, M, D, L	Тип данных ARRAY, STRUCT или PLC, в который сохраняются отдельные биты

Можно выбрать необходимую битовую последовательность из выпадающего списка "???" поля инструкций.

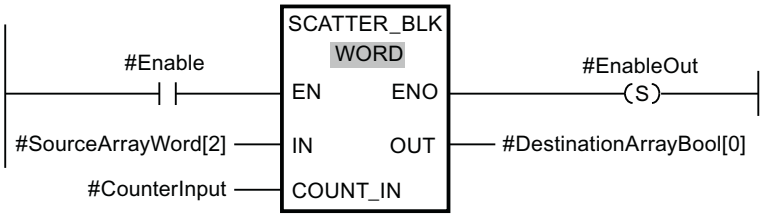
Дополнительную информацию о действующих типах данных можно найти в разделе "См. также".

Пример для целевого массива ARRAY с нижней границей "0"

Создать следующие переменные (теги) на интерфейсе блока:

Переменная (тег)	Область	Тип данных
Enable	Вход	BOOL
SourceArrayWord		ARRAY[0..5] из WORD
CounterInput		UDINT
EnableOut	Выход	BOOL
DestinationArrayBool		ARRAY[0..95] из BOOL

В следующем примере показан принцип работы данной инструкции:



В нижеследующей таблице показано, как инструкция работает со специальными значениями операндов:

Параметр	Операнд	Тип данных
IN	SourceArrayWord[2]	ARRAY[0..5] из WORD (может быть проанализировано 96 битов)
COUNT_IN	CounterInput = 3	UDINT3 (3 x WORD или 48 бит должны быть проанализированы. Это значит, что в целевом массиве ARRAY должно быть доступно минимум 48 бит.)
OUT	DestinationArrayBool[0]	Операнд "DestinationArrayBool" имеет тип данных ARRAY[0..95] из BOOL. Это значит, что он предоставляет 96 элементов BOOL.

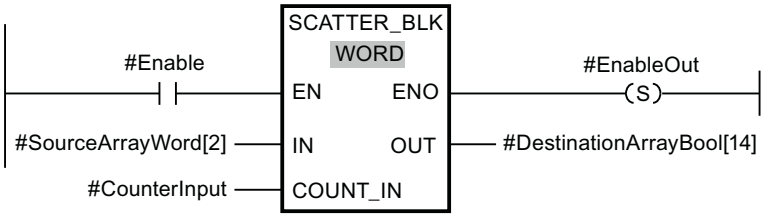
Если операнд " #Enable" возвращает на разрешающем входе EN состояние сигнала "1", то инструкция выполняется. Третье, четверное и пятое слово WORD операнда #SourceArrayWord разбивается на отдельные биты (48) и назначается отдельным элементам операнда #DestinationArrayBool, начиная с первого элемента. Если при выполнении инструкции возникает ошибка, то операнд #EnableOut выводит на разрешающий выход ENO состояние сигнала "0".

Пример для целевого массива ARRAY с нижней границей "-2"

Создать следующие переменные (теги) на интерфейсе блока:

Переменная (тег)	Область	Тип данных
Enable	Вход	BOOL
SourceArrayWord		ARRAY[0..5] из WORD
CounterInput		UDINT
EnableOut	Выход	BOOL
DestinationArrayBool		ARRAY[-2..93] из BOOL

В следующем примере показан принцип работы данной инструкции:



В нижеследующей таблице показано, как инструкция работает со специальными значениями операндов:

Параметр	Операнд	Тип данных
IN	SourceArrayWord[2]	ARRAY[0..5] из WORD (может быть проанализировано 96 битов)
COUNT_IN	CounterInput = 3	UDINT3 (3 x WORD или 48 бит должны быть проанализированы. Это значит, что в целевом массиве ARRAY должно быть доступно минимум 48 бит.)
OUT	DestinationArrayBool[14]	Операнд "DestinationArrayBool" имеет тип данных ARRAY[-2..93] из BOOL. Это значит, что он предоставляет 96 элементов BOOL.

Если операнд #Enable возвращает на разрешающем входе EN состояние сигнала "1", то инструкция выполняется. Третье, четверное и пятое слово WORD операнда #SourceArrayWord разбивается на отдельные биты (48) и начиная с 16-го элемента назначается отдельным элементам операнда #DestinationArrayBool. Если при выполнении инструкции возникает ошибка, то операнд #EnableOut выводит на разрешающий выход ENO состояние сигнала "0". Оставшиеся 32 бита не записываются.

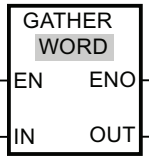
Дополнительную информацию и программный код для приведенного выше примера можно найти здесь: Библиотека образцов для инструкций.

8.6.13 GATHER

GATHER

Инструкция Сборка отдельных битов в несколько элементов массива ARRAY битовой последовательности соединяет биты типа данных "массив ARRAY", типа данных "anonyme STRUCT" или типа данных PLC только с булевыми элементами в битовую последовательность. Битовая последовательность сохраняется в переменную (тег) с типом данных BYTE, WORD, DWORD или LWORD.

Таблица 8-104 GATHER

LAD/FBD	SCL	Описание
	`GATHER(IN := #SourceArray, OUT => #DestinationArray);`	GATHER: Инструкция Сборка отдельных битов в несколько элементов массива ARRAY битовой последовательности соединяет биты типа данных "массив ARRAY", типа данных "anonyme STRUCT" или типа данных PLC только с булевыми элементами в битовую последовательность. Битовая последовательность сохраняется в переменную (тег) с типом данных BYTE, WORD или DWORD.

Примечание

Многомерный массив ARRAY из BOOL

В инструкции Сборка отдельных битов в несколько элементов массива ARRAY битовой последовательности нельзя использовать многомерный массив ARRAY из BOOL.

Примечание

Длина типа данных ARRAY, STRUCT или PLC

Число элементов в типе данных ARRAY, STRUCT или PLC должно точно соответствовать указанному в битовой последовательности числу элементов.

Для типа данных BYTE это означает, к примеру, что тип данных ARRAY, anonyme STRUCT или PLC должен иметь точно 8 элементов (WORD = 16 и DWORD = 32).

Примечание

Доступность инструкции

Инструкция может использоваться с CPU серии S7-1200 от версии прошивки 4.2 и с CPU серии S7-1500 с версией прошивки 2.1.

Разрешающий выход ENO возвращает состояние сигнала "0", если имеет место одно из следующих условий:

- Разрешающий вход EN имеет состояние сигнала "0".
- Число элементов BOOL в типе данных ARRAY, anonyme STRUCT или PLC не соответствует указанному в битовой последовательности числу элементов. В этом случае элементы BOOL не передаются.
- Имеется меньше бит, чем необходимо.

Типы данных для инструкции GATHER

В приведенной ниже таблице показаны параметры инструкции:

Параметр	Объявление	Тип данных		Область памяти	Описание
		S7-1200	S7-1500		
EN	Input	BOOL	BOOL	I, Q, M, D, L или постоянная	Разрешающий вход
ENO	Output	BOOL	BOOL	I, Q, M, D, L	Разрешающий выход

Параметр	Объявление	Тип данных		Область памяти	Описание
		S7-1200	S7-1500		
IN	Input	Тип данных ARRAY[*] из BOOL, STRUCT или PLC *: 8, 16, 32 или 64 элемента	Тип данных ARRAY[*] из BOOL, STRUCT или PLC *: 8, 16, 32 или 64 элемента	I, Q, M, D, L	Тип данных ARRAY, STRUCT или PLC, биты которого будут объединены в битовую последовательность. Значения не могут находиться в области ввода-вывода или в DB технологического объекта.
OUT	Output	BYTE, WORD, DWORD	BYTE, WORD, DWORD	I, Q, M, D, L	Собранная битовая последовательность, сохраненная в переменную (тег)

Можно выбрать необходимую битовую последовательность из выпадающего списка "???" поля инструкций.

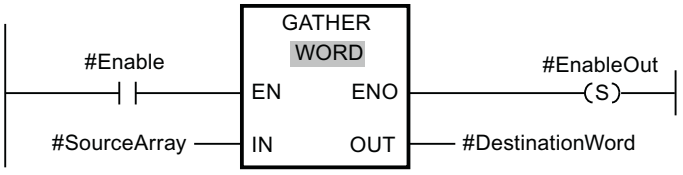
Дополнительную информацию о действующих типах данных можно найти в разделе "См. также".

Пример с массивом ARRAY

Создать следующие переменные (теги) на интерфейсе блока:

Переменная (тег)	Область	Тип данных
Enable	Вход	BOOL
SourceArray		ARRAY[0..15] из BOOL
EnableOut	Выход	BOOL
DestinationWord		WORD

В следующем примере показан принцип работы данной инструкции:



В нижеследующей таблице показано, как инструкция работает со специальными значениями операндов:

Параметр	Операнд	Тип данных
IN	SourceArray	Операнд "SourceArray" имеет тип данных ARRAY[0..15] из BOOL. Он состоит из 16 элементов и тем самым имеет точно такой же размер, что и тип данных WORD, в который должны быть собраны биты.
OUT	DestinationWord	WORD (16 бит)

Если операнд #Enable возвращает на разрешающем входе EN состояние сигнала "1", то инструкция выполняется. Биты операнда #SourceArray объединяются в WORD. Если при выполнении инструкции возникает ошибка, то операнд #EnableOut выводит на разрешающий выход ENO состояние сигнала "0".

Дополнительную информацию и программный код для приведенного выше примера можно найти здесь: Библиотека образцов для инструкций.

Пример с типом данных PLC (UDT)

Создать следующий блок данных PLC "myBits":

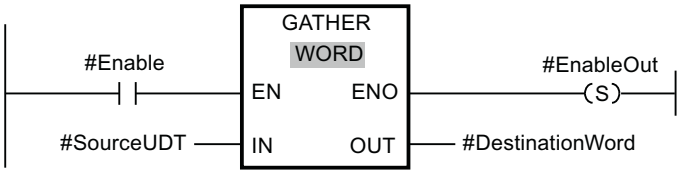
myBits

	Name	Data type
1	GeneralError	Bool
2	DeviceError	Bool
3	CommError	Bool
4	myError1	Bool
5	myError2	Bool
6	myError3	Bool
7	myError4	Bool
8	myError5	Bool
9	myError6	Bool
10	myError7	Bool
11	myError8	Bool
12	myError9	Bool
13	myError10	Bool
14	myError11	Bool
15	myError12	Bool
16	myError13	Bool

Создать следующие переменные (теги) на интерфейсе блока:

Переменная (тег)	Область	Тип данных
Enable	Вход	BOOL
SourceUDT		"myBits"
EnableOut	Выход	BOOL
DestinationWord		WORD

В следующем примере показан принцип работы данной инструкции:



В нижеследующей таблице показано, как инструкция работает со специальными значениями операндов:

Параметр	Операнд	Тип данных
IN	SourceUDT	Операнд "SourceUDT" имеет тип данных PLC (UDT). Он состоит из 16 элементов и тем самым имеет точно такой же размер, что и тип данных WORD, в который должны быть собраны биты.
OUT	DestinationWord	WORD (16 бит)

Если операнд #Enable возвращает на разрешающем входе EN состояние сигнала "1", то инструкция выполняется. Биты операнда #SourceUDT объединяются в WORD. Если при выполнении инструкции возникает ошибка, то операнд #EnableOut выводит на разрешающий выход ENO состояние сигнала "0".

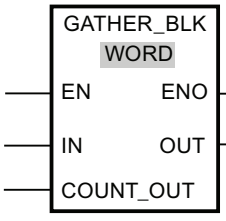
8.6.14 GATHER_BLK

Описание

Инструкция Сборка отдельных битов в несколько элементов массива ARRAY битовой последовательности соединяет биты типа данных "массив ARRAY", типа данных "anonyme STRUCT" или типа данных PLC только с булевыми элементами в один или несколько элементов массива ARRAY из <битовой последовательности>. В параметре COUNT_OUT указывается, сколько элементов целевого массива ARRAY должно быть записано. На этом этапе также не явно указывается, сколько потребуется элементов типа данных ARRAY из BOOL, anonyme STRUCT или PLC. Целевой массив ARRAY в параметре OUT может иметь больше элементов, чем указано в параметре COUNT_OUT. Массив ARRAY из <битовой последовательности> должен иметь достаточно элементов для сохранения собираемых битов. Но и целевой массив ARRAY может иметь больший размер.

Таблица 8-105 GATHER

LAD/FBD	SCL	Описание

	`GATHER_BLK(IN := #SourceArrayBool[0],`
	`COUNT_OUT := #CounterOutput,`
	`OUT => #DestinationArrayWord[2]);`
		Инструкция Сборка отдельных битов в несколько элементов массива ARRAY битовой последовательности соединяет биты типа данных "массив ARRAY", типа данных "anonyme STRUCT" или типа данных PLC только с булевыми элементами в один или несколько элементов массива ARRAY из <битовой последовательности>. В параметре COUNT_OUT указывается, сколько элементов целевого массива ARRAY должно быть записано. На этом этапе также не явно указывается, сколько потребуется элементов типа данных ARRAY из BOOL, anonyme STRUCT или PLC. Целевой массив ARRAY в параметре OUT может иметь больше элементов, чем указано в параметре COUNT_OUT. Массив ARRAY из <битовой последовательности> должен иметь достаточно элементов для сохранения собираемых битов. Но и целевой массив ARRAY может иметь больший размер.

Примечание

Данные НЕ записываются, если ENO = FALSE

У S7-1200 CPU только для инструкции GATHER_BLK действует правило, что данных не записываются на выход, если ENO = FALSE.

Примечание

Многомерный массив ARRAY из BOOL

Если в случае ARRAY речь идет о многомерном массиве из BOOL, то при подсчете учитываются и биты заполнения размерностей, даже если они не были явно объявлены.

Пример 1: Массив ARRAY[1..10,0..4,1..2] из BOOL обрабатывается как ARRAY[1..10,0..4,1..8] из BOOL или как ARRAY[0..399] из BOOL.

Пример 2: На параметре OUT подключается ARRAY[0..5] из WORD (sourceArrayWord[2]). Параметр COUNT_IN имеет значение "3". На параметре IN подключается ARRAY[0..1,0..5,0..7] из BOOL (destinationArrayBool[0,0,0]). Как массив на параметре IN, так и массив на параметре OUT, имеют размер в 96 бит. 48 отдельных битов включаются в сборку из ARRAY из BOOL.

Примечание

Если нижняя граница массива ARRAY исходного массива ARRAY не "0", то необходимо учитывать следующее:

Для поддержания производительности индекс всегда должен начинаться на границе BYTE, WORD или DWORD. Это значит, что расчет индекса должен начинаться на нижней границе массива ARRAY. За основу для этого расчета берется следующая формула:

Правильные индексы = Нижняя граница массива ARRAY + n(число битовых последовательностей) × число битов в нужной битовой последовательности

Для массива ARRAY[-2..45] из BOOL и битовой последовательности WORD расчет выглядит следующим образом:

- Правильный индекс (-2) = -2 + 0 × 16
- Правильный индекс (14) = -2 + 1 × 16
- Правильный индекс (30) = -2 + 2 × 16

Пример представлен ниже.

Примечание

Доступность инструкции

Инструкция может использоваться с CPU серии S7-1200 от версии прошивки 4.2 и с CPU серии S7-1500 с версией прошивки 2.1.

Разрешающий выход ENO возвращает состояние сигнала "0", если имеет место одно из следующих условий:

- Разрешающий вход EN имеет состояние сигнала "0".
- Индекс исходного массива ARRAY не начинается на границе BYTE, WORD или DWORD. В этом случае результат не записывается в массив ARRAY из <битовой последовательности>.
- Тип данных ARRAY[*] из <битовой последовательности> не располагает необходимым числом элементов.
  - S7-1500 CPU: В этом случае выполняется объединение максимально возможного числа битовых последовательностей с записью в тип данных ARRAY из <битовой последовательности>. Прочие биты в дальнейшем не учитываются.
  - S7-1200 CPU: Копирование не выполняется.

Типы данных для инструкции GATHER_BLK

В приведенной ниже таблице показаны параметры инструкции:

Параметр	Объявление	Тип данных		Область памяти	Описание
		S7-1200	S7-1500		
EN	Input	BOOL	BOOL	I, Q, M, D, L или постоянная	Разрешающий вход
ENO	Output	BOOL	BOOL	I, Q, M, D, L	Разрешающий выход
IN	Input	Элемент типа данных ARRAY[*] из BOOL, STRUCT или PLC	Элемент типа данных ARRAY[*] из BOOL, STRUCT или PLC	I, Q, M, D, L	Тип данных ARRAY из BOOL, STRUCT или PLC, биты которого объединяются (исходный ARRAY) Значения не могут находиться в области ввода-вывода или в DB технологического объекта.

Параметр	Объявление	Тип данных		Область памяти	Описание
		S7-1200	S7-1500		
COUNT_OUT	Input	USINT, UINT, UDINT	USINT, UINT, UDINT	I, Q, M, D, L	Счетчик, в какое число элементов целевого массива ARRAY должна быть выполнена запись. Значение не может находиться в области ввода-вывода или в базе данных технологического объекта.
OUT	Output	Элемент массива ARRAY[*] из <битовой последовательности>	Элемент массива ARRAY[*] из <битовой последовательности>	I, Q, M, D, L	ARRAY из <битовой последовательности>, в который будут сохранены биты (целевой ARRAY)

Можно выбрать необходимую битовую последовательность из выпадающего списка "???" поля инструкций.

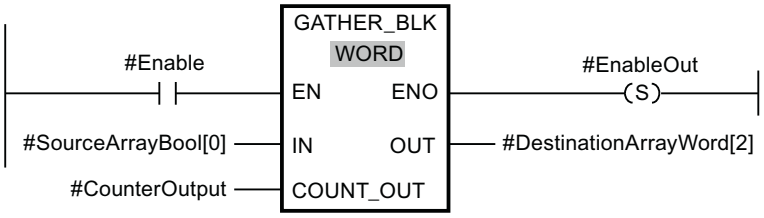
Дополнительную информацию о действующих типах данных можно найти в разделе "См. также".

Пример для исходного массива ARRAY с нижней границей "0"

Создать следующие переменные (теги) на интерфейсе блока:

Переменная (тег)	Область	Тип данных
Enable	Вход	BOOL
SourceArrayBool		ARRAY[0..95] из BOOL
CounterOutput		UDINT
EnableOut	Выход	BOOL
DestinationArrayWord		ARRAY[0..5] из WORD

В следующем примере показан принцип работы данной инструкции:



В нижеследующей таблице показано, как инструкция работает со специальными значениями операндов:

Параметр	Операнд	Тип данных
IN	SourceArrayBool[0]	Операнд "SourceArrayBool" имеет тип данных ARRAY[0..95] из BOOL. Это значит, что он предоставляет 96 элементов BOOL, которые снова могут быть собраны в слова.
COUNT_OUT	CounterOutput = 3	UDINT3 (3 слова должны быть записаны. Это значит, что в исходном массиве ARRAY должно быть доступно 48 бит.)
OUT	DestinationArrayWord[2]	Операнд "DestinationArrayWord" относится к типу данных ARRAY[0..5] из WORD. Это значит, что доступно 6 элементов WORD.

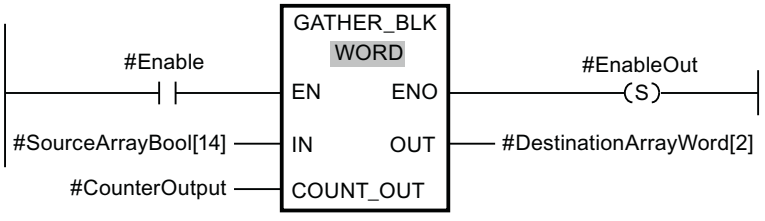
Если операнд #Enable возвращает на разрешающем входе EN состояние сигнала "1", то инструкция выполняется. Начиная с первого элемента операнда #SourceArrayBool, 48 бит объединяются в операнде #DestinationArrayWord. Начальной точкой в целевом массиве ARRAY является третий элемент. Это означает, что первые 16 бит будут записаны в третье слово, вторые 16 бит - в четвертое слово и третьи 16 бит - в пятое слово в целевом массиве ARRAY. Если при выполнении инструкции возникает ошибка, то операнд #EnableOut выводит на разрешающий выход ENO состояние сигнала "0".

Пример для исходного массива ARRAY с нижней границей "-2"

Создать следующие переменные (теги) на интерфейсе блока:

Переменная (тег)	Область	Тип данных
Enable	Вход	BOOL
SourceArrayBool		ARRAY[-2..93] из BOOL
CounterOutput		UDINT
EnableOut	Выход	BOOL
DestinationArrayWord		ARRAY[0..5] из WORD

В следующем примере показан принцип работы данной инструкции:



В нижеследующей таблице показано, как инструкция работает со специальными значениями операндов:

Параметр	Операнд	Тип данных
IN	SourceArrayBool[14]	Операнд "SourceArrayBool" имеет тип данных ARRAY[-2..93] из BOOL. Так как начальной точкой является 16-й элемент, доступно только 80 элементов BOOL, которые снова могут быть собраны в слова.
COUNT_OUT	CounterOutput = 3	UDINT3 (3 слова должны быть записаны. Это значит, что в исходном массиве ARRAY должно быть доступно 48 бит.)
OUT	DestinationArrayWord[2]	Операнд "DestinationArrayWord" относится к типу данных ARRAY[0..5] из WORD. Это значит, что доступно 6 элементов WORD.

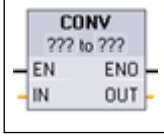
Если операнд #Enable возвращает на разрешающем входе EN состояние сигнала "1", то инструкция выполняется. Начиная с 16-го элемента операнда #SourceArrayBool, 48 бит объединяются в операнд #DestinationArrayWord. Начальной точкой в целевом массиве ARRAY является третий элемент. Это значит, что первые 16 бит исходного массива ARRAY будут проигнорированы. Вторые 16 бит будут записаны в третье слово, третьи 16 бит - в четвертое слово и четвертые 16 бит - в пятое слово в целевом массиве ARRAY. Оставшиеся 64 бита исходного массива ARRAY также не будут учтены. Если при выполнении инструкции возникает ошибка, то операнд #EnableOut выводит на разрешающий выход ENO состояние сигнала "0".

Дополнительную информацию и программный код для приведенного выше примера можно найти здесь: Библиотека образцов для инструкций.

8.7 Операции преобразования

8.7.1 CONV (преобразование значения)

Таблица 8-106 Инструкция преобразования (CONV)

LAD/FBD	SCL	Описание
	`out := <data type in>_TO_<data type out>(in);`	Преобразует элемент данных из одного типа данных в другой тип данных.

[1] В LAD и FBD: Нажать на "???" и выбрать типы данных из выпадающего списка.

[2] В SCL: Для создания инструкции преобразования следует указать один тип для входного параметра (in) и другой тип для выходного параметр (out). Пример: DWORD_TO_REAL преобразует значение DWord в Real.

Таблица 8-107 Типы данных для параметров

Параметр	Тип данных	Описание
IN	Битовая строка[1], SInt, USInt, Int, UInt, DInt, UDInt, Real, LReal, BCD16, BCD32, Char, WChar	Входное значение
OUT	Битовая строка[1], SInt, USInt, Int, UInt, DInt, UDInt, Real, LReal, BCD16, BCD32, Char, WChar	Входное значение, преобразованное в новый тип данных

[1] Для инструкции нельзя выбирать битовые строки (Byte, Word, DWord). Для ввода для параметра инструкции операнда с типом данных Byte, Word, или DWord, следует выбрать целое число без знака с той же длиной в битах. Пример: Выбрать USInt для Byte, UInt для Word и UDInt для DWord.

После выбора исходного типа данных (преобразовать из), в выпадающем списке предлагается ряд вариантов для преобразования (преобразовать в). Преобразования из и в BCD16 ограничиваются типом данных Int. Преобразования из и в BCD32 ограничиваются типом данных DInt.

Таблица 8-108 Состояние ENO

ENO	Описание	Результат OUT
1	Ошибки отсутствуют	Правильный результат
0	IN = +/- INF или +/- NaN	+/- INF или +/- NaN
0	Результат превышает допустимый диапазон для типа данных OUT	OUT устанавливается на значение из IN

8.7.2 Инструкции преобразования в SCL

Инструкции преобразования в SCL

Таблица 8-109 Преобразование из Bool, Byte, Word или DWord

Тип данных	Инструкция	Результат
Bool	**BOOL_TO_BYTE, BOOL_TO_WORD, BOOL_TO_DWORD, BOOL_TO_INT, BOOL_TO_DINT**	Значение передается в младший бит целевого типа данных.
Byte	**BYTE_TO_BOOL**	Младший бит передается в целевой тип данных.
	BYTE_TO_WORD, BYTE_TO_DWORD	Значение передается в младший байт целевого типа данных.
	BYTE_TO_SINT, BYTE_TO_USINT	Значение передается в целевой тип данных.
	BYTE_TO_INT, BYTE_TO_UINT, BYTE_TO_DINT, BYTE_TO_UDINT	Значение передается в младший байт целевого типа данных.
Word	**WORD_TO_BOOL**	Младший бит передается в целевой тип данных.
	WORD_TO_BYTE	В целевой тип данных передается младший байт исходного значения.
	WORD_TO_DWORD	Значение передается в младшее слово целевого типа данных.
	WORD_TO_SINT, WORD_TO_USINT	В целевой тип данных передается младший байт исходного значения.
	WORD_TO_INT, WORD_TO_UINT	Значение передается в целевой тип данных.
	WORD_TO_DINT, WORD_TO_UDINT	Значение передается в младшее слово целевого типа данных.
DWord	**DWORD_TO_BOOL**	Младший бит передается в целевой тип данных.
	DWORD_TO_BYTE, DWORD_TO_WORD, DWORD_TO_SINT	В целевой тип данных передается младший байт исходного значения.
	DWORD_TO_USINT, DWORD_TO_INT, DWORD_TO_UINT	В целевой тип данных передается младшее слово исходного значения.
	DWORD_TO_DINT, DWORD_TO_UDINT, DWORD_TO_REAL	Значение передается в целевой тип данных.

Таблица 8-110 Преобразование короткого целого числа (SInt или USInt)

Тип данных	Инструкция	Результат
SInt	**SINT_TO_BOOL**	Младший бит передается в целевой тип данных.
	SINT_TO_BYTE	Значение передается в целевой тип данных.
	SINT_TO_WORD, SINT_TO_DWORD	Значение передается в младший байт целевого типа данных.
	SINT_TO_INT, SINT_TO_DINT, SINT_TO_USINT, SINT_TO_UINT, SINT_TO_UDINT, SINT_TO_REAL, SINT_TO_LREAL, SINT_TO_CHAR, SINT_TO_STRING	Значение будет преобразовано.
USInt	**USINT_TO_BOOL**	Младший бит передается в целевой тип данных.
	USINT_TO_BYTE	Значение передается в целевой тип данных.
	USINT_TO_WORD, USINT_TO_DWORD, USINT_TO_INT, USINT_TO_UINT, USINT_TO_DINT, USINT_TO_UDINT	Значение передается в младший байт целевого типа данных.
	USINT_TO_SINT, USINT_TO_REAL, USINT_TO_LREAL, USINT_TO_CHAR, USINT_TO_STRING	Значение будет преобразовано.

Таблица 8-111 Преобразование целого числа (Int или UInt)

Тип данных	Инструкция	Результат
Int	**INT_TO_BOOL**	Младший бит передается в целевой тип данных.
	INT_TO_BYTE, INT_TO_DWORD, INT_TO_SINT, INT_TO_USINT, INT_TO_UINT, INT_TO_UDINT, INT_TO_REAL, INT_TO_LREAL, INT_TO_CHAR, INT_TO_STRING	Значение будет преобразовано.
	INT_TO_WORD	Значение передается в целевой тип данных.
	INT_TO_DINT	Значение передается в младший байт целевого типа данных.
UInt	**UINT_TO_BOOL**	Младший бит передается в целевой тип данных.
	UINT_TO_BYTE, UINT_TO_SINT, UINT_TO_USINT, UINT_TO_INT, UINT_TO_REAL, UINT_TO_LREAL, UINT_TO_CHAR, UINT_TO_STRING	Значение будет преобразовано.
	UINT_TO_WORD, UINT_TO_DATE	Значение передается в целевой тип данных.
	UINT_TO_DWORD, UINT_TO_DINT, UINT_TO_UDINT	Значение передается в младший байт целевого типа данных.

Таблица 8-112 Преобразование двойного целого числа (Dint или UDInt)

Тип данных	Инструкция	Результат
DInt	**DINT_TO_BOOL**	Младший бит передается в целевой тип данных.
	DINT_TO_BYTE, DINT_TO_WORD, DINT_TO_SINT, DINT_TO_USINT, DINT_TO_INT, DINT_TO_UINT, DINT_TO_UDINT, DINT_TO_REAL, DINT_TO_LREAL, DINT_TO_CHAR, DINT_TO_STRING	Значение будет преобразовано.
	DINT_TO_DWORD, DINT_TO_TIME	Значение передается в целевой тип данных.
UDInt	**UDINT_TO_BOOL**	Младший бит передается в целевой тип данных.
	UDINT_TO_BYTE, UDINT_TO_WORD, UDINT_TO_SINT, UDINT_TO_USINT, UDINT_TO_INT, UDINT_TO_UINT, UDINT_TO_DINT, UDINT_TO_REAL, UDINT_TO_LREAL, UDINT_TO_CHAR, UDINT_TO_STRING	Значение будет преобразовано.
	UDINT_TO_DWORD, UDINT_TO_TOD	Значение передается в целевой тип данных.

Таблица 8-113 Преобразование вещественного числа (Real или LReal)

Тип данных	Инструкция	Результат
Real	**REAL_TO_DWORD, REAL_TO_LREAL**	Значение передается в целевой тип данных.
	REAL_TO_SINT, REAL_TO_USINT, REAL_TO_INT, REAL_TO_UINT, REAL_TO_DINT, REAL_TO_UDINT, REAL_TO_STRING	Значение будет преобразовано.
LReal	**LREAL_TO_SINT, LREAL_TO_USINT, LREAL_TO_INT, LREAL_TO_UINT, LREAL_TO_DINT, LREAL_TO_UDINT, LREAL_TO_REAL, LREAL_TO_STRING**	Значение будет преобразовано.

Таблица 8-114 Преобразование из Time, DTL, TOD или Date

Тип данных	Инструкция	Результат
Time	**TIME_TO_DINT**	Значение передается в целевой тип данных.
DTL	**DTL_TO_DATE, DTL_TO_TOD**	Значение будет преобразовано.
TOD	**TOD_TO_UDINT**	Значение будет преобразовано.
Date	**DATE_TO_UINT**	Значение будет преобразовано.

Таблица 8-115 Преобразование из Char или String

Тип данных	Инструкция	Результат
Char	**CHAR_TO_SINT, CHAR_TO_USINT, CHAR_TO_INT, CHAR_TO_UINT, CHAR_TO_DINT, CHAR_TO_UDINT**	Значение будет преобразовано.
	CHAR_TO_STRING	Значение передается в первый символ строки.
String	**STRING_TO_SINT, STRING_TO_USINT, STRING_TO_INT, STRING_TO_UINT, STRING_TO_DINT, STRING_TO_UDINT, STRING_TO_REAL, STRING_TO_LREAL**	Значение будет преобразовано.
	STRING_TO_CHAR	Первый символ строки копируется в Char.

8.7.3 Инструкции ROUND (округление числа) и TRUNC (округление до целого числа)

Таблица 8-116 Инструкции ROUND und TRUNC

LAD/FBD	SCL	Описание
	**out := ROUND (in);**	преобразует вещественное число в целое. В LAD/FBD нажать на "???" в поле инструкции, чтобы выбрать тип данных для выхода, например, "DInt". В SCL типом данных для выхода инструкции ROUND по умолчанию является DINT. Для округления в другой тип выходных данных ввести имя инструкции с явным именем типа данных, например, ROUND_REAL или ROUND_LREAL. Дробная часть вещественного числа округляется до ближайшего целого значения (IEEE - округление до ближайшего). Если дробная часть находится ровно посередине между двумя целыми (например, 10,5), то результат округляется до большего целого. Пример: • ROUND (10.5) = 10 • ROUND (11.5) = 12
	**out := TRUNC(in);**	TRUNC преобразует вещественное число в целое. Дробная часть вещественного числа отбрасывается (IEEE - округление до нуля).

[1] В LAD и FBD: Нажать на "???" (для имени инструкции) и выбрать тип данных из выпадающего списка.

Таблица 8-117 Типы данных для параметров

Параметр	Тип данных	Описание
IN	Real, LReal	Вход для числа с плавающей запятой
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Округленный или целочисленный выход

Таблица 8-118 Состояние ENO

ENO	Описание	Результат OUT
1	Ошибки отсутствуют	Правильный результат
0	IN = +/- INF или +/- NaN	+/- INF или +/- NaN

8.7.4 CEIL и FLOOR (округление числа с плавающей запятой до следующего целого числа в сторону увеличения или уменьшения)

Таблица 8-119 Инструкции CEIL und FLOOR

LAD/FBD	SCL	Описание
	`out := CEIL(in);`	Преобразует вещественное число (Real или LReal) в целое число большее или равное выбранному вещественному числу (IEEE - округление до +бесконечности).
	`out := FLOOR(in);`	Преобразует вещественное число (Real или LReal) в целое число меньшее или равное выбранному вещественному числу (IEEE - округление до -бесконечности).

[1] В LAD и FBD: Нажать на "???" (для имени инструкции) и выбрать тип данных из выпадающего списка.

Таблица 8-120 Типы данных для параметров

Параметр	Тип данных	Описание
IN	Real, LReal	Вход для числа с плавающей запятой
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Преобразованный выход

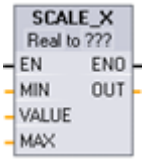
Таблица 8-121 Состояние ENO

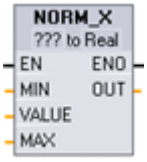
ENO	Описание	Результат OUT
1	Ошибки отсутствуют	Правильный результат
0	IN = +/- INF или +/- NaN	+/- INF или +/- NaN

8.7.5 SCALE_X (масштабирование) и NORM_X (нормализация)

Таблица 8-122 Инструкции SCALE_X и NORM_X

LAD/FBD	SCL	Описание

	`out :=SCALE_X(min:=_in_,`
	`value:=_in_,`
	`max:=_in_);`
		Масштабирует нормализованный вещественный параметр VALUE (0,0 <= VALUE <= 1,0) согласно типу данных и диапазону значений, определенному параметрами MIN и MAX:
		OUT = VALUE (MAX - MIN) + MIN

	`out :=NORM_X(min:=_in_,`
	`value:=_in_,`
	`max:=_in_);`
		Нормализует параметр VALUE к диапазону значений, определенному параметрами MIN и MAX:
		OUT = (VALUE - MIN) / (MAX - MIN),
		где (0,0 <= OUT <= 1,0)

[1] В LAD и FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.

Таблица 8-123 Типы данных для параметров

Параметр	Тип данных[1]	Описание
MIN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Вход для минимального значения диапазона
VALUE	SCALE_X: Real, LReal NORM_X: SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Входное значение для масштабирования или нормализации
MAX	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Вход для максимального значения диапазона
OUT	SCALE_X: SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal NORM_X: Real, LReal	Масштабированное или нормализованное выходное значение

[1] Для SCALE_X: Параметры MIN, MAX и OUT должны иметь одинаковый тип данных.
Для NORM_X: Параметры MIN, VALUE и MAX должны иметь одинаковый тип данных.

Примечание

Параметр VALUE изSCALE_X должен лежать в диапазоне (0,0 <= VALUE <= 1,0).

Если параметр VALUE меньше чем 0,0 или больше чем 1,0, то:

- Операция линейного масштабирования может создавать значения на выходе OUT, которые меньше, чем значение параметра MIN или больше, чем значение параметра MAX, если речь идет о значениях OUT, которые лежат в диапазоне значений типа данных OUT. В таких случаях при выполнении SCALE_X устанавливается ENO = TRUE.
- Возможно генерирование масштабированных чисел, которые не находятся в диапазоне типа данных OUT. В этом случае значение параметра OUT устанавливается на промежуточное значение, соответствующее младшей значимой части масштабированного вещественного числа перед окончательным преобразованием в тип данных OUT. В таких случаях при выполнении SCALE_X устанавливается ENO = FALSE.

Параметр VALUE изNORM_X должен лежать в диапазоне (MIN <= VALUE <= MAX).

Если параметр VALUE меньше, чем MIN, или больше, чем MAX, то инструкция линейного масштабирования может генерировать нормализованные значения на выходе OUT, меньшие 0,0 или большие 1,0. В этом случае исполнение NORM_X устанавливает ENO = TRUE.

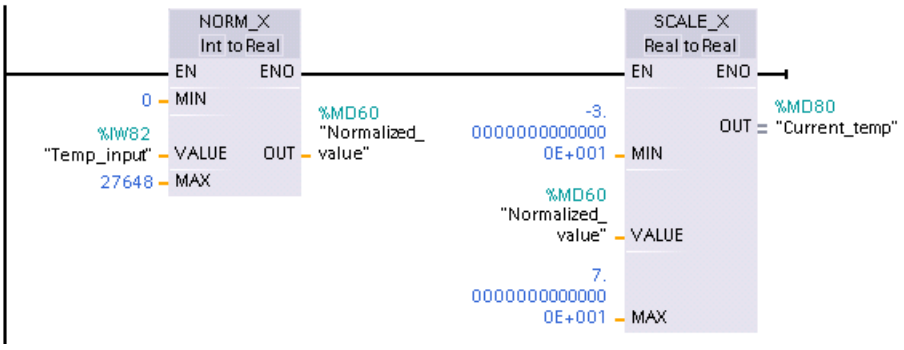
Таблица 8-124 Состояние ENO

ENO	Условие	Результат OUT
1	Ошибки отсутствуют	Правильный результат
0	Результат превышает допустимый диапазон для типа данных OUT.	Промежуточный результат: Младшая значащая часть вещественного числа до заключительного преобразования в тип данных OUT.
0	ПараметрMAX <= MIN	SCALE_X: Младшая значащая часть вещественного числа VALUE, занимающая место OUT. NORM_X: Значение VALUE в типе данных VALUE, расширенное для заполнения участка памяти размером в двойное слово.
0	Параметр VALUE = +/- INF или +/- NaN	VALUE записывается OUT

Пример (LAD): нормализация и масштабирование аналогового входного значения

Аналоговый вход аналогового сигнального модуля или сигнальной платы с входом по току лежит в диапазоне от 0 до 27648 для правильных значений. Предположим, что аналоговый вход представляет температуру, где значение 0 аналогового входа соответствует -30,0 градусам C, а значение 27648 соответствует температуре в 70,0 градусов C.

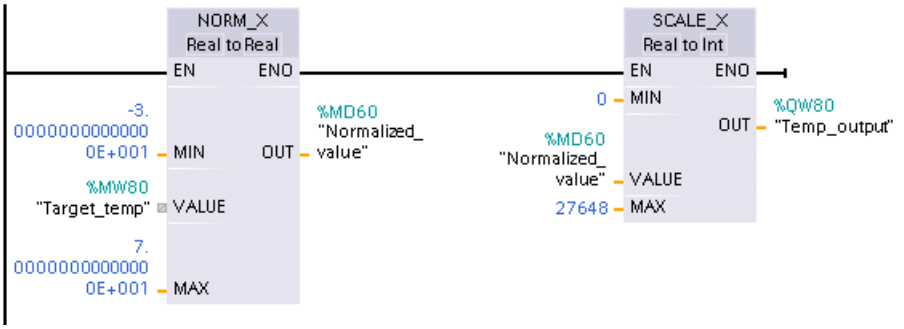
Для преобразования аналогового значения в физические соответствующие единицы, следует нормализовать вход в значения от 0,0 до 1,0, а затем масштабировать в диапазон от -30,0 до 70,0. Результирующее значение является температурой, представленной аналоговым входом в градусах C:



Следует учитывать, что если аналоговый вход использует напряжение от аналогового сигнального модуля или сигнальной платы с входом по напряжению, то значение MIN для инструкции NORM_X будет равно -27648 вместо 0.

Пример (LAD): Нормализация и масштабирование аналогового выходного значения

Аналоговый выход, устанавливаемый в аналоговом сигнальном модуле или сигнальной плате с выходом по току, должен лежать в диапазоне от 0 до 27648 для правильных значений. Предположим, что аналоговый выход представляет установку температуры, где значение 0 аналогового входа соответствует -30,0 градусам C, а значение 27648 соответствует температуре в 70,0 градусов C. Чтобы преобразовать значение температуры в памяти, находящееся между -30,0 и 70,0, в значение для аналогового выхода в диапазоне от 0 до 27648, необходимо нормализовать значение в физических единицах в значение от 0,0 до 1,0, а затем масштабировать его в диапазон аналогового выхода от 0 до 27648:



Следует учитывать, что если аналоговый выход предусмотрен для аналогового сигнального модуля или сигнальной платы с входом по напряжению, то значение MIN для инструкции SCALE_X будет равно -27648 вместо 0.

Дополнительную информацию о представлении аналоговых входов (Страница 1548) и о представлении аналоговых выходов (Страница 1549) для напряжения и тока можно найти в технических данных.

8.7.6 Инструкции преобразования типа Variant

8.7.6.1 VARIANT_TO_DB_ANY (преобразование из VARIANT в DB_ANY)

С помощью инструкции "VARIANT to DB_ANY" можно прочесть операнд из параметра IN с последующим его преобразованием в тип данных DB_ANY. Параметр IN имеет тип данных Variant представляет собой либо блок данных экземпляра, либо блок данных ARRAY. При создании программы не важно, какой блок данных соответствует параметру IN. Инструкция считывает номер блока данных при работе и записывает его в операнд параметра RET_VAL .

Таблица 8-125 Инструкция VARIANT_TO_DB_ANY

LAD/FBD	SCL	Описание
Недоступно	`RET_VAL := VARIANT_TO_DB_ANY( in := _variant_in_, err => _int_out_);`	Считывает операнд из параметра Variant IN и сохраняет его в результат функции типа DB_ANY

Таблица 8-126 Параметры для инструкции VARIANT_TO_DB_ANY

Параметр	Тип данных	Описание
IN	Variant	Элемент Variant, который представляет блок данных экземпляра или блок данных массива.
RET_VAL	DB_ANY	Выводимый тип данных DB_ANY, который содержит преобразованный номер блока данных.
ERR	Int	Информация об ошибке

Таблица 8-127 Состояние ENO

ENO	Условие	Результат
1	Ошибки отсутствуют	Инструкция преобразует вход Variant и сохраняет его на выходе функции DB_ANY.
0	Разрешающий вход EN имеет состояние сигнала "0" или параметр IN является недействительным.	Инструкция не работает.

Таблица 8-128 Код ошибки для инструкции VARIANT_TO_DB_ANY

Err (W#16#...)	**Описание**
0000	Ошибки отсутствуют
252C	Тип данных Variant из параметра IN имеет значение 0. CPU переходит в рабочее состояние STOP.
8131	Блок данных не существует или слишком короткий (первое обращение).
8132	Блок данных слишком короткий и не является блоком данных Array (второе обращение).
8134	Блок данных защищен от записи.
8150	Тип данных Variant из параметра IN посылает значение "0". Для данного сообщения об ошибке необходимо активировать свойство блока "Обработка ошибок в блоке". В противном случае CPU перейдет в режим STOP и передаст код ошибки 16#252C
8154	Неверный тип данных у блока данных.
*Коды ошибок могут отображаться в редакторе тестов программ в виде целых или шестнадцатеричных чисел.	

8.7.6.2 DB_ANY_TO_VARIANT (преобразование DB_Any в VARIANT)

С помощью инструкции "DB_ANY to VARIANT" может быть считан номер блока данных, отвечающий следующим условиям. Операнд в параметре IN имеет тип данных DB_ANY, поэтому при создания программы не обязательно знать, какой блок данных должен быть считан. Инструкция считывает номер блока данных при работе и записывает его с помощью указателя VARIANTв результат функции RET_VAL.

Таблица 8-129 Инструкция DB_ANY_TO_VARIANT

LAD/FBD	**SCL**	**Описание**
Недоступно	`RET_VAL := DB_ANY_TO_VARIANT( in := _db_any_in_, err => _int_out_);`	Считывает номер блока данных из параметра Variant IN и сохраняет его в результат функции типа VARIANT.

Таблица 8-130 Параметры для инструкции DB_ANY_TO_VARIANT

Параметр	**Тип данных**	**Описание**
IN	DB_ANY	Элемент Variant, содержащий номер блока данных
RET_VAL	Variant	Выводимый тип данных DB_ANY, который содержит преобразованный номер блока данных.
ERR	Int	Информация об ошибке

Таблица 8-131 Состояние ENO

ENO	Условие	Результат
1	Ошибки отсутствуют	Инструкция преобразует номер блока данных в формат Variant и сохраняет его на выход функции DB_ANY
0	Разрешающий вход EN имеет состояние сигнала "0" или параметр IN является недействительным.	Инструкция не работает.

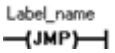
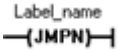
Таблица 8-132 Код ошибки для инструкции DB_ANY_TO_VARIANT

Err (W#16#...)	Описание
0000	Ошибки отсутствуют
8130	Номер блока данных равен 0.
8131	Блок данных не существует или слишком короткий.
8132	Блок данных слишком короткий и не является блоком данных массива.
8134	Блок данных защищен от записи.
8154	Неверный тип данных у блока данных.
8155	Неизвестный код типа
*Коды ошибок могут отображаться в редакторе тестов программ в виде целых или шестнадцатеричных чисел.	

8.8 Управление программой

8.8.1 Инструкции JMP (переход при RLO = 1), JMPN (переход при RLO = 0) и Label (метка перехода)

Таблица 8-133 Инструкции JMP, JMPN и LABEL

LAD	FBD	SCL	Описание
		См. инструкцию GOTO (Страница 333).	Переход при RLO (результат логической операции) = 1: При наличии потока сигналов к катушке JMP (LAD) или если вход блока JMP (FBD) имеет значение TRUE, тогда обработка программы продолжается с первой инструкции, следующей за указанной меткой перехода.
			Переход при RLO = 0: При отсутствии потока сигналов к катушке JMP (LAD) или если вход блока JMP (FBD) имеет значение FALSE, тогда обработка программы продолжается с первой инструкции, следующей за указанной меткой перехода.
			Цель перехода для инструкции перехода JMP или JMPN

[1] Создание имен для меток перехода осуществляется путем прямого ввода инструкции LABEL. С помощью символа параметра можно выбирать доступные метки перехода для инструкции JMP или JMPN. Также можно ввести имя метки перехода непосредственно в инструкцию JMP или JMPN.

Таблица 8-134 Типы данных для параметров

Параметр	Тип данных	Описание
Label_name	Идентификатор метки перехода	Идентификатор для инструкций перехода и соответствующая метка перехода для цели перехода.

- Каждая метка перехода должна быть уникальной в своем блоке кода.
- Можно выполнять переходы внутри одного блока кода, но переход в другие блоки невозможен.
- Переход может выполняться как вперед так и назад.
- Можно перейти к одной и той же метке более чем из одного места в одном и том же блоке кода.

8.8.2 JMP_LIST (определение списка переходов)

Таблица 8-135 Инструкция JMP_LIST

LAD/FBD	SCL	Описание

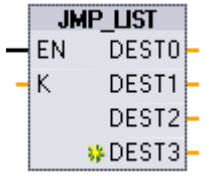
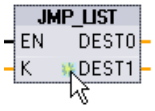
	`CASE k OF`
	`0: GOTO dest0;`
	`1: GOTO dest1;`
	`2: GOTO dest2;`
	`[n: GOTO destn;]`
	`END_CASE;`
		Инструкция JMP_LIST распределяет переходы в программе для управления обработкой ее сегментов. В зависимости от значения на входе K выполняется переход на соответствующую метку перехода в программе. Выполнение программы продолжается с инструкций, следующих за меткой перехода. Если значение на входе K превышает число меток перехода - 1, то переход не выполняется и обработка продолжается со следующего сегмента в программе.

Таблица 8-136 Типы данных для параметров

Параметр	Тип данных	Описание
K	UInt	Управляющее значение для распределения переходов
DEST0, DEST1, .., DESTn.	Метки перехода в программе	Метки перехода целей перехода согласно значениям параметра K: Если значение K равно 0, то выполняется переход на метку перехода, которая назначена для выхода DEST0. Если значение K равно 1, то выполняется переход на метку перехода, которая назначена для выхода DEST1 и т. д. Если значение на входе K превышает число меток перехода - 1, то переход не выполняется и обработка продолжается со следующего сегмента в программе.

В LAD и FBD: При вставке блока JMP_LIST в программу имеется два выхода для меток перехода. Можно добавлять и удалять цели перехода.



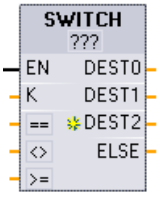
Нажать в блоке на символ создания (слева от последнего параметра DEST) для добавления новых выходов для меток перехода.



- Кликнуть правой кнопкой мыши по выходному соединителю и выбрать "Вставить вход".
- Кликнуть правой кнопкой мыши по выходному соединителю и выбрать "Удалить".

8.8.3 SWITCH (распределение переходов)

Таблица 8-137 Инструкция SWITCH

LAD/FBD	SCL	Описание
	**Недоступно**	Инструкция SWITCH распределяет переходы в программе для управления обработкой ее сегментов. В зависимости от результатов сравнения между значением на входе K и значениями на указанных сравнительных входах, происходит переход к метке, соответствующей первому результату сравнения с результатом TRUE. Если ни одно сравнение не является истинным, то выполняется переход к метке, которой назначен ELSE. Выполнение программы продолжается с инструкций, следующих за меткой перехода.

[1] В LAD и FBD: Кликнуть под именем блока и выбрать из выпадающего списка тип данных.

[2] В SCL: Следует использовать сравнения через IF-THEN.

Таблица 8-138 Типы данных для параметров

Параметр	Тип данных[1]	Описание
K	UInt	Вход для общего заданного значения
==, <>, <, <=, >. >=	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, TOD, Date	Входы для отдельных заданных значений для специальных типов сравнения
DEST0, DEST1, .., DESTn, ELSE	Метки перехода в программе	Метки перехода целей перехода согласно специальным сравнениям: Сравнительны вход под и рядом с входом K обрабатывается вначале и вызывает переход к метке, назначенной DEST0, если сравнение между значением K и этим входом истинно. Следующая сравнительная проверка использует следующий вход ниже и вызывает переход к метке, назначенной DEST1, если сравнение истинно. Оставшиеся сравнения обрабатываются схожим образом, и, если ни одно из сравнений не истинно, то происходит переход к метке, назначенной выходу ELSE.

[1] Вход K и сравнительные входы (==, <>, <, <=, >, >=) должны иметь один и тот же тип данных.

Добавление, удаление входов и указание типов сравнения

При вставке блока SWITCH в LAD или FBD в программе имеется два сравнительных входа. Можно назначать типы сравнения и добавлять и удалять входы или цели перехода (см. ниже).



Кликнуть в блоке по оператору сравнения и выбрать из выпадающего списка нового оператора.



Кликнуть в блоке по символу создания (слева от последнего параметра DEST) для добавления новых параметров для целей сравнения.



- Кликнуть правой кнопкой мыши по входному соединителю и выбрать "Вставить вход".
- Кликнуть правой кнопкой мыши по входному соединителю и выбрать "Удалить".

Таблица 8-139 Выбор типа данных для блока SWITCH и допустимые операции сравнения

Тип данных	Сравнение	Синтаксис оператора
Byte, Word, DWord	Равно	==
	Не равно	<>
SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, TOD, Date	Равно	==
	Не равно	<>
	Больше или равно	>=
	Меньше или равно	<=
	Больше чем	>
	Меньше чем	<

Правила для размещения блока SWITCH

- Перед сравнительным входом не должно быть подключенных инструкций LAD/FBD.
- Выход ENO отсутствует, поэтому разрешена только одна инструкция SWITCH в сегменте, и инструкций SWITCH должна быть последней инструкцией в сегменте.

8.8.4 RET (возвращаемое значение)

Опциональная инструкция RET используется для завершения исполнения текущего блока. Только если есть поток сигналов к катушке RET (LAD) или если вход блока RET имеет значение TRUE (FBD), то выполнение программы в текущем блоке будет завершено в данном месте; следующие за RET инструкции выполнены не будут. Если текущим блоком является OB, то параметр "Return_Value" будет проигнорирован. Если текущим блоком является FC или FB, то значение параметра "Return_Value" передается обратно в вызывающую процедуру в качестве значения ENO вызванного блока.

Не требуется вставлять инструкцию RET в качестве последней инструкции блока, это производится автоматически. Можно вставить несколько инструкций RET в один блок.

Für SCL см. инструкцию RETURN (Страница 333).

Таблица 8-140 Инструкция управления программой "Возвращаемое значение" (RET)

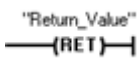
LAD	FBD	SCL	Описание
		`RETURN;`	Завершает исполнение текущего блока.

Таблица 8-141 Типы данных для параметров

Параметр	Тип данных	Описание
Return_Value	Bool	Параметр "Return_value" инструкции RET назначается на выход ENO инструкции вызова в вызывающем блоке.

Примеры для использования инструкции RET в блоке кода FC:

1. Создать новый проект и вставить FC:
2. Отредактировать FC:
   - Добавить инструкции из каталога инструкций.
   - Вставить инструкцию RET, включая один из следующих элементов для параметра "Return_Value":
     TRUE, FALSE или адрес, указывающий требуемое возвращаемое значение.
   - Добавить другие инструкции.
3. Вызвать FC из главной программы MAIN [OB1].

Вход EN блока FC в блоке кода MAIN должен быть TRUE, чтобы выполнить FC.

Значение, указанное инструкцией RET в FC, будет присутствовать на выходе ENO блока FC в блоке кода MAIN после выполнения FC, для которого есть прохождение сигнала к инструкции RET.

8.8.5 ENDIS_PW (включить/выключить пароль CPU)

Таблица 8-142 Инструкция ENDIS_PW

LAD/FBD	SCL	Описание

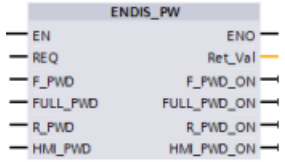
	`ENDIS_PW(`
	`  req:=_bool_in_,`
	`  f_pwd:=_bool_in_,`
	`  full_pwd:=_bool_in_,`
	`  r_pwd:=_bool_in_,`
	`  hmi_pwd:=_bool_in_,`
	`  f_pwd_on=>_bool_out_,`
	`  full_pwd_on=>_bool_out_,`
	`  r_pwd_on=>_bool_out_,`
	`  hmi_pwd_on=>_bool_out_);`
		С помощью инструкции ENDIS_PW можно разрешить или запретить клиентские соединения с S7-1200 CPU, даже при корректном вводе пароля клиентом.
		Данная инструкция не запрещает пароли к веб-серверу.

Таблица 8-143 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Выполнение функции, если REQ=1
F_PWD	IN	Bool	Пароль для F-режима: Разрешить (=1) или блокировать (=0)
FULL_PWD	IN	Bool	Пароль на полный доступ: Разрешить пароль для полного доступа (=1) или заблокировать (=0)
R_PWD	IN	Bool	Пароль на доступ по чтению: Разрешить (=1) или блокировать (=0)
HMI_PWD	IN	Bool	Пароль для HMI: Разрешить (=1) или блокировать (=0)
F_PWD_ON	OUT	Bool	Состояние пароля для F-режима: Разрешен (=1) или заблокирован (=0)
FULL_PWD_ON	OUT	Bool	Состояние пароля для полного доступа: Разрешен (=1) или заблокирован (=0)
R_PWD_ON	OUT	Bool	Состояние пароля для доступа по чтению: Разрешен (=1) или заблокирован (=0)
HMI_PWD_ON	OUT	Bool	Состояние пароля для HMI: Разрешен (=1) или заблокирован (=0)
Ret_Val	OUT	Word	Результат функции

Вызов ENDIS_PW с REQ=1 запрещает использование типов паролей, у которых соответствующий входной параметр имеет состояние FALSE. Каждый тип пароля может быть разрешен или запрещен по отдельности. Например, если разрешается только fail-safe пароль, а все остальные пароли запрещены, то можно запретить доступ к CPU для определенной небольшой группы пользователей.

ENDIS_PW выполняется синхронно в цикле программы, и выходные параметры пароля всегда показывают текущее состояние разрешения пароля, независимого от входного параметра REQ. Все разрешаемые пароли должны иметь и опцию для их запрещения. Иначе выводится ошибка и разрешаются все пароли, которые были разрешены до выполнения ENDIS_PW. Это означает, что в стандартном CPU (где fail-safe пароль не

сконфигурирован), F_PWD должен всегда устанавливаться в 1, чтобы дать возвращаемое значение 0. В этом случае F_PWD_ON всегда равен 1.

Примечание

- Выполнение ENDIS_PW может заблокировать доступ к устройствам HMI, если пароль HMI заблокирован.
- Клиентские сеансы, которые были разрешены до выполнения ENDIS_PW, остаются неизменными после выполнения ENDIS_PW.

После запуска доступ к CPU ограничен с помощью паролей, назначенных в обычной конфигурации защиты CPU. Возможность блокировки действующего пароля должна быть заново установлена через повторное выполнение инструкции ENDIS_PW. Если же ENDIS_PW выполняется сразу же и требуемые пароли из-за этого блокируются, то доступ в TIA Portal может быть заблокирован. Можно воспользоваться таймерной инструкцией для установки задержки выполнения ENDIS_PW и предоставить время для ввода пароля, до того как они будут заблокированы.

Примечание

Восстановление CPU, коммуникация которого с TIA Portal была заблокирована

См. информацию в разделе "Восстановление потерянного пароля (Страница 149)" чтобы узнать, как стереть внутреннюю загружаемую память PLC при помощи карты памяти.

Изменение рабочего состояния на STOP, вызванное ошибками, выполнением STP или действиями STEP 7, не влияет на защиту. Защита работает до выключения и повторного включения CPU. См. следующую таблицу для получения более подробной информации.

Действие	Рабочее состояние	Защита паролем ENDIS_PW
После восстановления исходного состояния памяти через STEP 7	STOP	Активно: Заблокированные пароли остаются заблокированными.
После включения смены карты памяти	STOP	Выкл.: Не заблокированных паролей.
После выполнения ENDIS_PW в OB цикла или пусковом OB	STARTUP, RUN	Активно: Пароли заблокированы согласно параметрам ENDIS_PW
После смены режимы работы из RUN или STARTUP в STOP из-за инструкции STP, ошибки или действий в STEP 7	STOP	Активно: Заблокированные пароли остаются заблокированными

Примечание

Использовать сложные пароли для уровней доступа CPU. Сложные пароли - это, по крайней мере, десять символов, представляющих комбинацию букв, цифр и специальных символов, не являющиеся словами, которые могут быть найдены в словаре, и не являющиеся именами или идентификаторами, которые могут быть получены из персональных данных. Следует хранить пароль в секрете и часто его изменять.

Таблица 8-144 Коды условий

RET_VAL (W#16#...)	Описание
0000	Ошибки отсутствуют
8090	Инструкция не поддерживается.
80D0	Fail-safe пароль не сконфигурирован.
80D1	Пароль на чтение/запись не сконфигурирован.
80D2	Пароль на чтение не сконфигурирован.
80D3	Пароль для доступа через HMI не сконфигурирован.

8.8.6 RE_TRIGR (перезапуск времени контроля цикла)

Таблица 8-145 Инструкция RE_TRIGR

LAD/FBD	SCL	Описание
	`RE_TRIGR();`	С помощью RE_TRIGR (перезапуск времени контроля цикла) можно увеличить максимально допустимое время до создания сообщения об ошибке со стороны контроля времени цикла.

С помощью инструкции RE_TRIGR выполняется перезапуск времени контроля цикла при выполнении цикла. При этом происходит продление максимально допустимого времени цикла на время одного максимального цикла с момента последнего использования функции RE_TRIGR.

Примечание

До версии 2.2 прошивки S7-1200 CPU, функция RE_TRIGR должна была использоваться в OB программного цикла для продления времени цикла целевой системы без ограничений. ENO = FALSE и контроль времени цикла не сбрасывается, если RE_TRIGR вызывается из пускового OB, OB прерывания или OB ошибки.

От версии прошивки 2.2, функция RE_TRIGR может выполняться из любого OB (включая пусковой OB, OB прерывания или OB ошибки). Но время цикла целевой системы может быть продлено не более чем на 10 значений сконфигурированного максимального времени цикла.

Установка максимального времени цикла устройства PLC

Сконфигурировать значение для максимального времени цикла в конфигурации устройства во "Времени цикла".

Таблица 8-146 Значения времени цикла

Контроль времени цикла	Минимальное значение	Максимальное значение	По умолчанию
Максимальное время цикла	1 мс	6000 мс	150 мс

Контроль времени выполнения

Если максимальное время цикла истек до того, как была закончен цикл, то будет сгенерирована ошибка. Если в программе пользователя есть OB прерывания по ошибке времени (OB 80), то он выполняется CPU. Он может содержать программную логику с соответствующей реакцией.

Если в программе пользователя нет OB прерывания по ошибке времени, то первое условие тайм-аута игнорируется и CPU остается в рабочем состоянии RUN. При втором превышении максимального времени цикла в одном программном цикле (2 x макс. значение времени цикла), будет сгенерирована ошибка, и целевая система перейдет в режим STOP.

В рабочем состоянии STOP выполнение программы будет остановлено, в то время как системные коммуникации и диагностика CPU продолжат работать.

8.8.7 STP (завершение программы)

Таблица 8-147 Инструкция STP

LAD/FBD	SCL	Описание
	`STP();`	STP переводит CPU в рабочее состояние STOP. Когда CPU находится в режиме STOP, выполнение программы и обновление входов и выходов в образе процесса остановлены.

Дополнительную информацию см. в: Конфигурирование выходов для перехода из RUN в STOP (Страница 115)

Если EN = TRUE, то CPU перейдет в режим STOP, выполнение программы будет остановлено, при этом состояние ENO не имеет значения. В противном случае, EN = ENO = 0.

8.8.8 Инструкции GET_ERROR и GET_ERROR_ID (локальное получение ошибки локальное получение ID ошибки)

Инструкции GET_ERROR предоставляют информацию об ошибках при выполнении программных блоков. После включения инструкции GET_ERROR или GET_ERROR_ID в свой блок кода, можно обрабатывать ошибки программы в своем программном блоке.

GET_ERROR

Таблица 8-148 Инструкция GET_ERROR

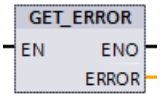
LAD/FBD	SCL	Описание
	`GET_ERROR(_out_);`	Указывает, что при выполнения программного блока произошла локальная ошибка и заполняет заранее определенную структуру данных подробной информацией об ошибке.

Таблица 8-149 Типы данных для параметров

Параметр	Тип данных	Описание
ERROR	ErrorStruct	Структура данных ошибки: Можно переименовать структуру, но не элементы в структуре

Таблица 8-150 Элементы структуры данных ErrorStruct

Компоненты структуры		Тип данных	Описание
ERROR_ID		Word	ID ошибки
FLAGS		Byte	Показывает, возникла ли при вызове блока ошибка. • 16#01: Ошибки при вызове блока. • 16#00: Отсутствие ошибок при вызове блока.
REACTION		Byte	Реакция по умолчанию: • 0: Игнорировать (ошибка записи), • 1: Продолжить с замещающим значением "0" (ошибка чтения), • 2: Пропустить инструкцию (системная ошибка)
CODE_ADDRESS		CREF	Информация об адресе и типе блока
	BLOCK_TYPE	Byte	Тип блока, в котором возникла ошибка: • 1: OB • 2: FC • 3: FB
	CB_NUMBER	UInt	Номер блока кода
	OFFSET	UDInt	Ссылка на внутреннюю память

Компоненты структуры		Тип данных	Описание					
MODE		Byte	Тип доступа: В зависимости от типа доступа может быть выведена следующая информация:					
			Режим работы	(A)	(B)	(C)	(D)	(E)
			0					
			1					Смещение
			2			Область		
			3	Адрес	Диапазон		Номер	
			4			Область		Смещение
			5			Область	Номер DB	Смещение
			6	№ указ./ акк.		Область	Номер DB	Смещение
			7	№ указ./ акк.	№ слота/ диап.	Область	Номер DB	Смещение
OPERAND_NUMBER		UInt	Номер операнда машинной команды					
POINTER_NUMBER_ LOCATION		UInt	(A) Внутренний указатель					
SLOT_NUMBER_SCOPE		UInt	(B) Область хранения во внутренней памяти					
DATA_ADDRESS		NREF	Информация об адресе операнда					
	AREA	Byte	(C) Область памяти: • L: 16#40 – 4E, 86, 87, 8E, 8F, C0 – CE • I: 16#81 • Q: 16#82 • M: 16#83 • DB: 16#84, 85, 8A, 8B					
	DB_NUMBER	UInt	(D) Номер блока данных					
	OFFSET	UDInt	(E) Относительный адрес операнда					

GET_ERROR_ID

Таблица 8-151 Инструкция GetErrorID

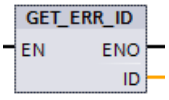
LAD/FBD	SCL	Описание
	`GET_ERR_ID();`	Указывает, что при выполнения программного блока произошла ошибка и сообщает ID (идентификатор) ошибки.

Таблица 8-152 Типы данных для параметров

Параметр	Тип данных	Описание
ID	Word	Идентификационные значения ошибки для ErrorStruct элемента ERROR_ID

Таблица 8-153 Значения Error_ID

ERROR_ID шест-надцатеричный	ERROR_ID десятич-ный	Ошибка выполнения блока программы
0	0	Ошибки отсутствуют
2520	9504	Повреждение строки
2522	9506	Ошибка чтения: операнд вне диапазона
2523	9507	Ошибка записи: операнд вне диапазона
2524	9508	Ошибка чтения: недействительная область
2525	9509	Ошибка записи: недействительная область
2528	9512	Ошибка чтения: выравнивание данных (неверное выравнивание бита)
2529	9513	Ошибка записи: выравнивание данных (неверное выравнивание бита)
252C	9516	Ошибка неинициализированного указателя
2530	9520	Блок данных защищен от записи
2533	9523	Недействительный указатель
2538	9528	Ошибка доступа: Блок данных отсутствует
2539	9529	Ошибка доступа: Неправильный DB
253A	9530	Глобальный DB отсутствует
253C	9532	Неверная версия или FC отсутствует
253D	9533	Инструкция отсутствует
253E	9534	Неверная версия или FB отсутствует
253F	9535	Инструкция отсутствует
2550	9552	Ошибка доступа: Блок данных отсутствует
2575	9589	Ошибка глубины вложенности программы
2576	9590	Ошибка размещения локальных данных
2942	10562	Физический вход отсутствует
2943	10563	Физический выход отсутствует

Принцип работы

По умолчанию CPU реагирует на ошибку выполнения блока, внося ошибку в буфер диагностики. Однако, если поместить одну или несколько инструкций GET_ERROR или GET_ERROR_ID в блок кода, то теперь сам этот блок теперь может обрабатывать ошибки в блоке. В этом случае CPU не вносит ошибку в буфер диагностики. Вместо этого информация об ошибке передается на выход инструкции GET_ERROR или GET_ERROR_ID. Для считывания подробной информацию об ошибке можно использовать инструкцию GET_ERROR, с помощью инструкции GET_ERROR_ID считывается только идентификатор ошибки. Обычно первая ошибка является самой важной, за которой идут ошибки, являющиеся ее следствием.

Первое выполнение инструкции GET_ERROR или GET_ERROR_ID в блоке возвращает первую ошибку, обнаруженную во время выполнения блока. Эта ошибка могла произойти где угодно между запуском блока и выполнением GET_ERROR или GET_ERROR_ID. Последующее выполнение GET_ERROR или GET_ERROR_ID возвращает первую ошибку, начиная с предыдущего выполнения GET_ERROR или GET_ERROR_ID. История ошибок не сохраняется, и выполнение инструкции повторно активирует целевая система, чтобы зафиксировать следующую ошибку.

Тип данных ErrorStruct, используемый инструкцией GET_ERROR, может быть добавлен в редактор блока данных и на интерфейсы блока, чтобы программа могла бы обращаться к этим значениям. Из выпадающего списка с типами данных, выбрать ErrorStruct, чтобы добавить данную структуру. Можно создать несколько элементов ErrorStruct, используя уникальные имена. Отдельные элементы ErrorStruct не могут быть переименованы.

Отображаемое ENO условие ошибки

Если: EN = TRUE и выполняются GET_ERROR или GET_ERROR_ID, то:

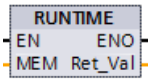
- ENO = TRUE означает, что возникла ошибка при выполнении блока кода и имеется информация об этой ошибке
- ENO = FALSE означает, что блок кода выполнен без ошибок

Можно связать логику программы обработки ошибки с входом ENO, который устанавливается после возникновения ошибки. При наличии ошибки ее данные хранятся в выходном параметре, где программа может к ним обращаться.

GET_ERROR иGET_ERROR_ID могут использоваться, чтобы отправить информацию об ошибке из выполняющегося в настоящий момент блока (вызываемый блок) в вызывающий блок. Поместить инструкцию в последний сегмент программы вызываемого блока, чтобы сообщить о заключительном состоянии выполнения вызываемого блока.

8.8.9 RUNTIME (измерение времени выполнения программы)

Таблица 8-154 Инструкция RUNTIME

LAD/FBD	SCL	Описание
	`Ret_Val := RUNTIME(` `  _lread_inout_);`	Производит измерение времени выполнения всей программы, отдельных блоков или последовательности команд.

Для измерения времени выполнения всей программы следует вызвать инструкцию "Измерение времени выполнения программы" в OB 1. Измерение времени выполнения начинается при первом вызове измеренное время выполнения программы выводится на выход RET_VAL после второго вызова. Измеренное время выполнения включает в себя все процессы CPU, которые могут возникнуть в процессе работы программы, например, прерывания, вызванные высокоуровневыми событиями или коммуникацией. Инструкция "Измерение времени выполнения программы" считывает внутренний счетчик CPU и записывает значение в IN-OUT параметр MEM. Инструкция

рассчитывает текущее время работы программы в соответствии с внутренней частотой счетчика и записывает это значение на выход RET_VAL.

Если необходимо измерить время работы отдельных блоков или последовательности команд, то потребуется три различных сегмента. Вызвать инструкцию "Измерение времени выполнения программы" в отдельном сегменте программы. С помощью первого вызова инструкции устанавливается начальная точка измерения времени работы. Затем вызвать в следующем сегменте требуемый программный блок или последовательность команд. Вызвать инструкцию "Измерение времени выполнения программы" в другом сегменте повторно и назначить IN-OUT параметруMEM ту же память, что и при первом вызове инструкции. Инструкция "Измерение времени выполнения программы" в третьем сегменте считывает значение внутреннего счетчика CPU, рассчитывает текущее время работы программных блоков или последовательности команд в соответствии с внутренней частотой счетчика и записывает это значение в выход RET_VAL.

Инструкция “Измерение времени выполнения программы" использует внутренний высокочастотный счетчик для вычисления времени. В случае переполнения счетчика инструкция возвращает значения <= 0,0. Можно проигнорировать данные значения.

Примечание

CPU не может точно определить время выполнения последовательности команд, поскольку порядок инструкций в командной последовательности изменяется во время оптимизированной компиляции программы.

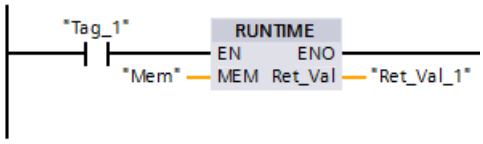
Таблица 8-155 Типы данных для параметров

Параметр	Тип данных	Описание
MEM	LReal	Начальная точка для измерения времени выполнения
RET_VAL	LReal	Измеренное время выполнения в секундах

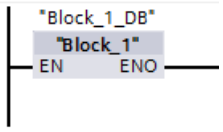
Пример: Инструкция RUNTIME

В следующем примере показано использование инструкции RUNTIME для измерения времени выполнения функционального блока:

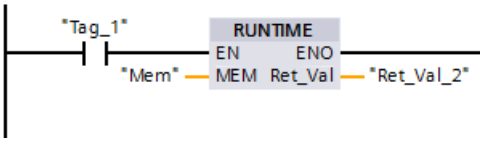
Сегмент 1:



Сегмент 2:



Сегмент 3:



Если операнд "Tag_1" в сети 1 имеет состояние сигнала "1", то выполняется инструкция RUNTIME. При первом вызове инструкции начальная точка измерения времени работы устанавливается и сохраняется в операнде "Mem" в качестве опорного значения для второго вызова инструкции.

Функциональный блок FB1 выполняется в сегменте 2.

Если программный блок FN1 выполнен и операнд "Tag_1" имеет состояние сигнала "1", то выполняется инструкция RUNTIME в сегменте 3. При втором вызове инструкции производится расчет времени выполнения программного блока и запись результата на выход RET_VAL_2.

8.8.10 Операторы управления программой в SCL

8.8.10.1 Обзор операторов управления программой в SCL

Структурированный язык управления SCL предлагает три типа операторов управления программой для структурирования программы пользователя:

- Операторы выбора: С помощью оператора выбора можно задать другую последовательность для обработки операторов в программе.
- Циклы: Можно управлять выполнением цикла, используя операторы цикла. Оператор цикла определяет, какие части программы должны быть выполнены повторно в зависимости от определенных условий.
- Переходы в программе: Переход в программе инициирует немедленное перемещение на указанное место назначения и следовательно к другому оператору в том же блоке.

Операторы управления программой используют синтаксис языка программирования PASCAL.

Таблица 8-156 Типы операторов управления программой в SCL

Операторы управления программой в SCL		**Описание**
Выбор	Оператор IF-THEN (Страница 326)	Позволяет направить выполнению программы в одно из двух альтернативных ветвлений, в зависимости от условия, имеющего состояние TRUE или FALSE.
	Оператор CASE (Страница 327)	Обеспечивает выборочное выполнение одного из *n* альтернативных ветвлений на основе значения переменной.
Цикл	Оператор FOR (Страница 329)	Служит для повторения последовательности операторов до те пор, пока управляющая переменная остается в указанном диапазоне значений.
	Оператор WHILE-DO (Страница 330)	Служит для повторения последовательности операторов до те пор, пока действует условие выполнения.
	Оператор REPEAT-UNTIL (Страница 331)	Служит для повторения последовательности операторов до те пор, пока не будет выполнено условие отмены.
Переход в программе	Оператор CONTINUE (Страница 331)	Служит для отмены выполнения текущего прохода цикла.
	Оператор EXIT (Страница 332)	Служит для выхода из цикла в любой точке, независимо от того, удовлетворено условие отмены или нет.
	Оператор GOTO (Страница 333)	Вызывает мгновенный переход на указанную метку.
	Оператор RETURN (Страница 333)	Вызывает выход из выполняемого блока кода и возврат в вызывающий блок кода.

8.8.10.2 Оператор IF-THEN

Условный оператор IF-THEN управляет ходом программы, выполняя группу операторов на базе обработки значения Bool логического выражения. Можно использовать скобки для вложения или структурирования нескольких операторов IF-THEN.

Таблица 8-157 Элементы оператора IF-THEN

SCL	**Описание**
IF "Условие" THEN **Оператор_A;** **Оператор_B;** **Оператор_C;** **;**	Если "Условие" TRUE или 1, выполнить следующие операторы до оператора END_IF. Если "Условие" FALSE или 0, выполнить переход к оператору END_IF (если в программе отсутствуют опциональные операторы ELSIF или ELSE).
[ELSIF "Условие-n" THEN **Оператор_N;** **;]**	Опциональный оператор ELSEIF[1] предлагает дополнительные условия для обработки. Пример: Если "Условие" в операторе IF-THEN = FALSE, то программа обрабатывает "Условие-n". Если "Условие-n" = TRUE, то выполнить "Оператор_N".

SCL	Описание
[ELSE **Оператор_X;** **;]**	Опциональный оператор ELSE предлагает операторы, которые должны быть выполнены, если "условие" оператора IF-THEN = FALSE.
END_IF;	Оператор END_IF завершает оператор IF-THEN.

[1] В оператор IF-THEN может быть вставлено несколько операторов ELSIF.

Таблица 8-158 Переменные оператора IF-THEN

Переменные	Описание
"Условие"	Обязательный элемент. Логическое выражение либо TRUE (1), либо FALSE (0).
"Инструкция_A"	Опциональный элемент. Один или несколько операторов, которые должны быть выполнены, если "Условие" = TRUE.
"Условие-n"	Опциональный элемент. Логическое выражение для обработки опциональным оператором ELSIF.
"Оператор_N"	Опциональный элемент. Один или несколько операторов, которые должны быть выполнены, если "Условие-n" оператора ELSIF = TRUE.
"Оператор_X"	Опциональный элемент. Один или несколько операторов, которые должны быть выполнены, если "Условие" оператора IF-THEN = FALSE.

Оператор IF выполняется согласно следующим правилам:

- Выполняется первый ряд операторов, логическое выражение которых = TRUE. Остальные ряды операторов не выполняются.
- Если нет булева выражения со значением TRUE, то выполняется введенный через ELSE ряд операторов (или выход из ряда операторов, если ветвление ELSE отсутствует).
- Количество операторов ELSIF не ограничено.

Примечание

Преимуществом использования одного или нескольких ветвлений ELSIF перед рядом операторов IF заключается в том, что логические выражения, которые следуют за правильным выражением, более не обрабатываются. За счет этого можно сократить время выполнения программы.

8.8.10.3 Оператор CASE

Таблица 8-159 Элементы оператора CASE

SCL	Описание
CASE "Тестовое значение" OF **"Список значений": Оператор[; Оператор, ...]** **"Список значений": Оператор[; Оператор, ...]** **[ELSE** **Оператор Else[; Оператор Else, ...]]** **END_CASE;**	Оператор CASE выполняет одну из нескольких групп операторов в зависимости от значения выражения.

Таблица 8-160 Параметр

Параметр	Описание
"Тестовое значение"	Обязательный элемент. Любое числовое выражение с типом данных Int.
"Список значений"	Обязательный элемент. Отдельное значение или список разделенных запятой значений или диапазонов значений. (для указания диапазона значений используются две точки: 2..8). В следующем примере показаны различные варианты списков значений: 1: Оператор_A; 2, 4: Оператор_B; 3, 5..7,9: Оператор_C;
Оператор	Обязательный элемент. Один или несколько операторов, которые выполняются, если "Тестовое значение" соответствует значению в списке значений.
Оператор Else	Опциональный элемент. Один или несколько операторов, которые выполняются, если ни одно из значений "Списка значений" не имеет соответствий.

Оператор CASE выполняется согласно следующим правилам:

- Выражение тестового значения должно возвращать значение типа Int.
- При обработке оператора CASE программа проверяет, содержится ли значение выражения тестового значения в указанном списке значений. Если совпадение найдено, то выполняется назначенный списку компонент оператора.
- Если совпадение не найдено, то выполняется следующий за ELSE сегмент программы. Если ветвление ELSEотсутствует, то оператор не выполняется.

Пример: Вложенные операторы CASE

Операторы CASE могут быть вложенными. Для каждого вложенного оператора CASE нужен свой оператор END_CASE.

```
CASE "var1" OF
       1 : #var2 := 'A';
       2 : #var2 := 'B';
ELSE
       CASE "var3" OF

          65..90: #var2 := ,Прописная буква';
          97..122: #var2 := ,Строчная буква';
       ELSE

          #var2:= 'Специальный симовл';
       END_CASE;
END_CASE;
```

8.8.10.4 Оператор FOR

Таблица 8-161 Элементы оператора FOR

SCL	Описание
FOR "Управляющая переменная" := "Начало" TO "Конец" **[BY "Инкремент"] DO** **Оператор;** **;** **END_FOR;**	Оператор FOR служит для повторения последовательности операторов до те пор, пока управляющая переменная остается в указанном диапазоне значений. Определение цикла с FOR включает в себя указание начального и конечного значения. Оба значения должны совпадать по типу с управляющей переменной. Циклы FOR поддерживают вложенность. Оператор END_FOR относится к последнему выполненному оператору FOR.

Таблица 8-162 Параметр

Параметр	Описание
"Управляющая переменная"	Обязательный элемент. Целое число (Int илиDInt) как счетчик циклов.
"Начало"	Обязательный элемент. Простое выражение, указывающее начальное значение управляющих переменных.
"Конец"	Обязательный элемент. Простое выражение, указывающее конечное значение управляющих переменных.
"Инкремент"	Опциональный элемент. Величина, на которую "Управляющая переменная" изменяется после каждого цикла. Тип "Инкремента" совпадает с типом "Управляющей переменной". Если значение "Инкремента" не указано, то значение переменной после каждого прохождения цикла увеличивается на 1. Во время выполнения оператора FOR изменение "Инкремента" невозможно.

Оператор FOR выполняется следующим образом:

- При запуске цикла управляющая переменная устанавливается на начальное значение (инициализация) и при каждом прохождении цикла увеличивается (положительный инкремент) или уменьшается (отрицательный инкремент) до достижения конечного значения.
- После каждого прохождения цикла проверяется, выполнено ли условие (конечное значение достигнуто) или нет. Если условие завершения не выполнено, то последовательность операторов выполнятся повторно; в ином случае цикл завершается, и выполнение продолжается со следующего непосредственно за циклом оператора.

Правила оформления операторов FOR:

- Управляющая переменная может иметь только тип данных Int или DInt.
- Оператор BY [инкремент] может быть опущен. Если инкремент не указывается, то автоматически применяется инкремент +1.

Для завершения цикла независимо от состояния выражения "Условие" следует использовать оператор Оператор EXIT (Страница 332). Оператор EXIT выполняет оператор, который следует сразу же за оператором END_FOR.

С помощью инструкции Оператор CONTINUE (Страница 331) можно пропустить последующие операторы цикла FOR и продолжить цикл с проверки, выполнено ли условие для завершения.

8.8.10.5 Оператор WHILE-DO

Таблица 8-163 Оператор WHILE

SCL	Описание
WHILE "Условие" DO **Оператор;** **Оператор;** **...;** **END_WHILE;**	Оператор WHILE выполняет ряд операторов, пока заданное условие остается TRUE. Циклы WHILE поддерживают вложенность. Оператор END_WHILE относится к последнему выполненному оператору WHILE.

Таблица 8-164 Параметр

Параметр	Описание
"Условие"	Обязательный элемент. Логическое выражение, анализ которого дает TRUE или FALSE. (Условие "Ноль" обрабатывается как FALSE.)
Оператор	Опциональный элемент. Один или несколько операторов, которые выполняются, пока условие = TRUE.

Примечание

Оператор WHILE анализирует состояние "Условия", прежде чем операторы начнут выполняться. Для выполнения операторов независимо от состояния "Условия" минимум один раз, следует использовать оператор REPEAT (Страница 331).

Оператор WHILE выполняется согласно следующим правилам:

- Перед обработкой содержания цикла анализируется условие выполнения.
- Следующее за DO содержание цикла будет повторяться до тех пор, пока условие выполнения остается на значении TRUE.
- Если появляется значение FALSE, то цикл пропускается и выполняется следующий за циклом оператор.

Для завершения цикла независимо от состояния выражения "Условие" следует использовать Оператор EXIT (Страница 332). Оператор EXIT выполняет оператор, который следует сразу же за оператором END_WHILE.

С помощью оператора CONTINUE можно пропустить последующие операторы цикла WHILE и продолжить цикл с проверки, выполнено ли условие для завершения.

8.8.10.6 Оператор REPEAT-UNTIL

Таблица 8-165 Оператор REPEAT

SCL	Описание
REPEAT **Оператор;** **;** **UNTIL "Условие"** **END_REPEAT;**	Оператор REPEAT выполняет группу операторов до тех пор, пока заданное условие остается TRUE. Циклы REPEAT поддерживают вложенность. Оператор END_REPEAT всегда относится к последнему выполненному оператору REPEAT.

Таблица 8-166 Параметр

Параметр	Описание
Оператор	Опциональный элемент. Один или несколько операторов, которые выполняются, пока условие = TRUE.
"Условие"	Обязательный элемент. Одно или несколько выражений двух следующих типов: Числовое выражение или строковое выражение, анализ которого дает TRUE или FALSE. Условие "Ноль" обрабатывается как FALSE.

Примечание

Перед анализом состояния "Условия" оператор REPEAT выполняется операторы при первом прохождении цикла (даже если "Условие" = FALSE). Для проверки состояния "Условия" перед выполнением операторов, следует использовать оператор WHILE (Страница 330).

Для завершения цикла независимо от состояния выражения "Условие" следует использовать Оператор EXIT (Страница 332). Оператор EXIT выполняет оператор, который следует сразу же за оператором END_REPEAT.

С помощью Оператор CONTINUE (Страница 331) можно пропустить последующие операторы цикла REPEAT и продолжить цикл с проверки, выполнено ли условие для завершения.

8.8.10.7 Оператор CONTINUE

Таблица 8-167 Оператор CONTINUE

SCL	Описание
CONTINUE **Оператор;** **;**	С помощью оператора CONTINUE можно пропустить последующие операторы программного цикла (FOR, WHILE, REPEAT) и продолжить цикл с проверки, выполнено ли условие для завершения. Если это не так, то цикл продолжается.

Оператор CONTINUE выполняется согласно следующим правилам:

- Этот оператор завершает исполнение содержания цикла сразу же.
- В зависимости от того, выполнено ли условие для повторения цикла или нет, содержание цикла выполняется повторно или оператор цикла завершается и выполняется следующий за ним оператор.
- В операторе FOR управляющая переменная, следующая непосредственно за оператором CONTINUE, увеличивается на указанный инкремент.

Следует использовать оператор CONTINUE только в цикле. Во вложенный цикла оператор CONTINUE всегда относятся к циклу, в котором он находится. CONTINUE как правило используется в комбинации с оператором IF.

Если цикл должен быть завершен независимо от проверки завершения, следует использовать оператор EXIT.

Пример: CONTINUE оператор

В следующем блоке показан пример использования оператора CONTINUE, для того чтобы избежать ошибки "деления на 0", вычисляя процент:

```
FOR i := 0 TO 10 DO
IF Значение[i] = 0 THEN CONTINUE; END_IF;
     p := Часть/ Значение[i] * 100;
     s := INT_TO_STRING(p);
  Процент:= CONCAT(IN1:=s, IN2:="%");
END_FOR;
```

8.8.10.8 Оператор EXIT

Таблица 8-168 Оператор EXIT

SCL	Описание
`EXIT;`	Оператор EXIT служит для завершения цикла (FOR, WHILE или REPEAT) в любом месте и независимо от того, выполнено ли условие завершения.

Оператор EXIT выполняется согласно следующим правилам:

- Этот оператор вызывает немедленное завершение оператора цикла, связанного с оператором EXIT.
- Выполнение программы продолжается по завершении цикла (например, после END_FOR).

Следует использовать оператор EXIT внутри цикла. Во вложенных циклах оператор EXIT возвращает обработку на следующий более высокий уровень вложенности.

Пример: EXIT оператор

```
FOR i := 0 TO 10 DO
CASE Значение[i, 0] OF
        1..10: Значение [i, 1]:="A";
        11..40: Значение [i, 1]:="B";
        41..100: Значение [i, 1]:="C";
ELSE
EXIT;
```

```
END_CASE;
END_FOR;
```

8.8.10.9 Оператор GOTO

Таблица 8-169 Оператор GOTO

SCL	Описание

`GOTO JumpLabel;`
`Оператор;`
`... ;`
`JumpLabel: Оператор;`
	Оператор GOTO пропускает операторы путем перехода на метку в том же блоке.
	Метка перехода ("JumpLabel") и оператор GOTO должны находиться в одном блоке. Имя метки перехода в пределах одного блока должно присваиваться только один раз. Любая метка перехода может быть целью различных операторов GOTO.

Невозможно перейти в сегмент цикла (FOR, WHILE или REPEAT). Переход из цикла возможен.

Пример: GOTO оператор

В следующем примере: В зависимости от значения операнда "Значение переменной (тега)", выполнение программы возобновляется в месте, заданном соответствующей меткой перехода. Если "Значение переменной (тега)" равно 2, то программа возобновляет работу на метке перехода "MyLabel2", пропустив "MyLabel1".

```
CASE "Значение переменной (тега)" OF
1 : GOTO Моя метка1;
2 : GOTO Моя метка2;
ELSE GOTO Моя метка3;
END_CASE;
Моя метка1: "Переменная (тег)_1" := 1;
Моя метка2: "Переменная (тег)_2" := 1;
Моя метка3: "Переменная (тег)_4" := 1;
```

8.8.10.10 Оператор RETURN

Таблица 8-170 Оператор RETURN

SCL	Описание
`RETURN;`	Оператор RETURN безусловно завершает обрабатываемый в данный момент блок кода. Выполнение программы возвращается на вызывающий блок или к операционной системе (при завершении OB).

Пример: RETURN оператор

```
IF "Ошибка" <> 0 THEN
RETURN;
END_IF;
```

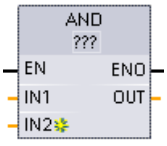
Примечание

После выполнения последнего оператора блок кода автоматический возвращается на вызывающий блок. Не вставлять оператор RETURN в конце блока кода.

8.9 Логические операции

8.9.1 Логические операции AND (И), OR (ИЛИ) и XOR (ИСКЛЮЧАЮЩЕЕ ИЛИ)

Таблица 8-171 Логические операции AND (И), OR (ИЛИ) и XOR (ИСКЛЮЧАЮЩЕЕ ИЛИ)

LAD/FBD	SCL	Описание
	`out := in1 AND in2;`	AND: Логическая операция И
	`out := in1 OR in2;`	OR: Логическая операция ИЛИ
	`out := in1 XOR in2;`	XOR: Логическая операция ИСКЛЮЧАЮЩЕЕ ИЛИ

[1] В LAD и FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.



Для добавления входа кликнуть по символу "Создать" или правой кнопкой мыши по одному из параметров IN на входном соединителе выбрать команду "Вставить вход".

Для удаления входа кликнуть правой кнопкой мыши на входном соединении одного из параметров IN (если имеется более двух исходных входов) и выбрать команду "Удалить".

Таблица 8-172 Типы данных для параметров

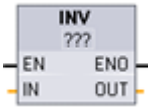
Параметр	Тип данных	Описание
IN1, IN2	Byte, Word, DWord	Логические входы
OUT	Byte, Word, DWord	Логический выход

[1] При выборе типа данных, параметры IN1, IN2 и OUT устанавливаются на один тип данных.

Соответствующие значения битов в IN1 и IN2 связываются в двоичный результат в параметре OUT. ENO после выполнения этих инструкций всегда имеет состояние TRUE.

8.9.2 INV (создание дополнения до единицы)

Таблица 8-173 Инструкция INV

LAD/FBD	SCL	Описание
	**Недоступно**	Рассчитывает двоичное дополнение до единицы для параметра IN. Дополнение до единицы образуется путем инвертирования каждого значения бита параметра IN (изменение 0 на 1 и 1 на 0). ENO после выполнения этой инструкции всегда имеет состояние TRUE.

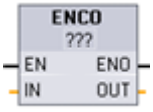
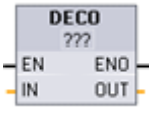
[1] В LAD и FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.

Таблица 8-174 Типы данных для параметров

Параметр	Тип данных	Описание
IN	SInt, Int, DInt, USInt, UInt, UDInt, Byte, Word, DWord	Элемент данных для инверсии
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Byte, Word, DWord	Инвертированное выходное значение

8.9.3 Инструкции DECO (декодирование) и ENCO (кодирование)

Таблица 8-175 Инструкции ENCO и DECO

LAD/FBD	SCL	Описание
	`out := ENCO(_in_);`	Преобразует битовую комбинацию в двоичное число. Инструкция ENCO преобразует параметр IN в двоичное число, соответствующее позиции двоичного разряда младшего бита параметра IN, и выводит результат в параметре OUT. Если параметр IN равен 0000 0001 или 0000 0000, то значение 0 возвращается в параметр OUT. Если значение параметра IN равно 0000 0000, то ENO устанавливается на FALSE.
	`out := DECO(_in_);`	Преобразует двоичное число в битовую комбинацию. Инструкция DECO преобразует двоичное число из параметра IN, устанавливая соответствующую двоичную позицию в параметре OUT на 1 (все другие биты устанавливаются на 0). ENO после выполнения инструкции DECO всегда имеет состояние TRUE. Примечание: Типом данных по умолчанию для инструкции DECO является DWORD. В SCL следует изменить имя инструкции на DECO_BYTE или DECO_WORD, чтобы декодировать значение Byte или Word с назначением переменной (тега) или адреса типа Byte или Word.

[1] В LAD и FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.

Таблица 8-176 Типы данных для параметров

Параметр	Тип данных	Описание
IN	ENCO: Byte, Word, DWord DECO: UInt	ENCO: Битовая комбинация для преобразования DECO: Двоичное значение для преобразования
OUT	ENCO: Int DECO: Byte, Word, DWord	ENCO: Преобразованное значение DECO: Преобразованная битовая комбинация

Таблица 8-177 Состояние ENO

ENO	Условие	Результат (OUT)
1	Ошибки отсутствуют	Правильный номер бита
0	IN равен нулю	OUT устанавливается на ноль.

Возможные типы данных Byte, Word илиDWord для параметра DECO OUT ограничивают полезный диапазон параметра IN. При нарушении значением параметра IN полезного диапазона, то выполняется операция по модулю для извлечения младших битов (см. ниже).

Область для параметра DECO IN:

- 3 бита (значения 0-7) IN используются для установки 1 позиции бита в Byte OUT
- 4 бита (значения 0-15) IN используются для установки 1 позиции бита в Word OUT
- 5 битов (значения 0-31) IN используются для установки 1 позиции бита в DWord OUT

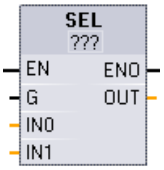
Таблица 8-178 Примеры

Значение IN для DECO			Значение OUT для DECO (преобразование отдельной позиции бита)
Byte OUT 8 бит	Мин. IN	0	00000001
	Макс. IN	7	10000000
Word OUT 16 бит	Мин. IN	0	0000000000000001
	Макс. IN	15	1000000000000000
DWord OUT 32 бит	Мин. IN	0	00000000000000000000000000000001
	Макс. IN	31	10000000000000000000000000000000

8.9.4 Инструкции SEL (выбор), MUX (мультиплексирование) и DEMUX (демультиплексирование)

Таблица 8-179 Инструкция SEL (выбор)

LAD/FBD	SCL	Описание

	`out := SEL(`
	`g:=_bool_in,`
	`in0:-_variant_in,`
	`in1:=_variant_in);`
		Команда SEL присваивает одно из двух входных значений параметру OUT, в зависимости от значения параметра G.

[1] В LAD и FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.

Таблица 8-180 Типы данных для инструкции SEL

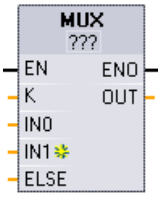
Параметр	Тип данных[1]	Описание
G	Bool	• 0 выбирает IN0 • 1 выбирает IN1
IN0, IN1	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Входы
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Выход

[1] Входные переменные и выходная переменная должны иметь одинаковый тип данных.

Коды условий: ENO после выполнения инструкции SEL всегда имеет состояние TRUE.

Таблица 8-181 Инструкция MUX (мультиплексирование)

LAD/FBD	SCL	Описание

	`out := MUX(`
	`k:=_unit_in,`
	`in1:=variant_in,`
	`in2:=variant_in,`
	
	`[...in32:=variant_in,]`
	`inelse:=variant_in);`
		Инструкция MUX копирует одно из нескольких входных значений, в зависимости от значения параметра K, в параметр OUT. Если значение параметра K превышает (IN*n* - 1), то значение параметра ELSE копируется в параметр OUT.

[1] В LAD и FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.



Для добавления входа кликнуть по символу "Создать" или правой кнопкой мыши по одному из параметров IN на входном соединителе выбрать команду "Вставить вход".

Для удаления входа кликнуть правой кнопкой мыши на входном соединении одного из параметров IN (если имеется более двух исходных входов) и выбрать команду "Удалить".

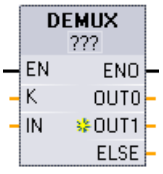
Таблица 8-182 Типы данных для инструкции MUX

Параметр	Тип данных	Описание
K	UInt	• 0 выбирает IN1 • 1 выбирает IN2 • *n* выбирает IN*n*
IN0, IN1, .. INn	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Входы
ELSE	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Замещающее значение для входа (опция)
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Выход

[1] Входные переменные и выходная переменная должны иметь одинаковый тип данных.

Таблица 8-183 Инструкция DEMUX (демультиплексирование)

LAD/FBD	SCL	Описание

	`DEMUX(`
	`  k:=_unit_in,`
	`  in:=variant_in,`
	`  out1:=variant_in,`
	`  out2:=variant_in,`
	
	`[...out32:=variant_in,]`
	
	`outelse:=variant_in);`
		Инструкция DEMUX копирует значение адреса, назначенного параметру IN, на один из нескольких выходов. Значение параметра K указывает, какой выход будет выбран в качестве цели для значения IN. Если значение K больше числа (OUT*n* - 1), то значение IN копируется в адрес, назначенный параметру ELSE.

[1] В LAD и FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.



Для добавления выхода кликнуть по символу "Создать" или правой кнопкой мыши по одному из параметров OUT на выходном соединителе и выбрать команду "Вставить выход".

Для удаления выхода кликнуть правой кнопкой мыши на входном соединении одного из параметров OUT (если имеется более двух исходных выходов) и выбрать команду "Удалить".

Таблица 8-184 Типы данных для инструкции DEMUX

Параметр	**Тип данных**[1]	**Описание**
K	UInt	Значение выбора: • 0 выбирает OUT1 • 1 выбирает OUT2 • n выбирает OUTn
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Вход
OUT0, OUT1, .. OUTn	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Выходы
ELSE	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Замещающее значение для выхода, если K больше чем (OUTn - 1)

[1] Входные переменные и выходные переменные должны иметь одинаковый тип данных.

Таблица 8-185 Состояние ENO для инструкций MUX и DEMUX

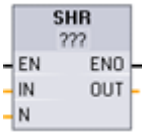
ENO	**Условие**	**Результат OUT**
1	Ошибки отсутствуют	MUX: Выбранное значение IN копируется в OUT DEMUX: ЗначениеIN копируется на выбранный OUT
0	MUX: K больше количества входов -1	• ELSE не указано: OUT не изменяется, • ELSE указано, значениеELSE присваивается OUT
	DEMUX: K больше количества выходов -1	• ELSE не указано: Выходы не изменяются, • ELSE указано, значениеIN копируется в ELSE

8.10 Сдвиг и циклический сдвиг

8.10.1 Инструкции SHR (сдвиг вправо) и SHL (сдвиг влево)

Таблица 8-186 Инструкции SHR и SHL

LAD/FBD	SCL	Описание

	`out := SHR(`
	`  in:=_variant_in_,`
	`  n:=_uint_in);`
	`out := SHL(`
	`  in:=_variant_in_,`
	`  n:=_uint_in);`
		С помощью инструкций сдвига (SHL и SHR) выполняется сдвиг битовой комбинации в параметре IN. Результат присваивается параметру OUT. Параметр N указывает количество сдвинутых позиций битов:
		• SHR: Сдвиг битовой комбинации вправо
		• SHL: Сдвиг битовой комбинации влево

[1] В LAD и FBD: Нажать на "???" и выбрать типы данных из выпадающего списка.

Таблица 8-187 Типы данных для параметров

Параметр	Тип данных	Описание
IN	Целые числа	Битовая комбинация для сдвига
N	USInt, UDint	Число позиций битов для сдвига
OUT	Целые числа	Битовая комбинация после сдвига

- При N=0 сдвиг не выполняется. Значение IN присваивается OUT.
- Битовые позиции, которые освобождаются операцией сдвига, заполняются нулями.
- Если число позиций для сдвига (N) превышает число битов в целевом значении (8 для Byte, 16 для Word, 32 дляDWord), то все первоначальные значения битов заменяются нулями (OUT присваивается ноль).
- Состояние ENO всегда TRUE при выполнении операций сдвига.

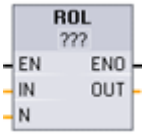
Таблица 8-188 Пример: SHL с данными Word

Сдвиг битов слова Word влево путем добавления нулей справа (N = 1)			
IN	1110 0010 1010 1101	Значение OUT до первого сдвига:	1110 0010 1010 1101
		После первого сдвига влево:	1100 0101 0101 1010
		После второго сдвига влево:	1000 1010 1011 0100
		После третьего сдвига влево:	0001 0101 0110 1000

8.10.2 Инструкции ROR (циклический сдвиг вправо) и ROL (циклический сдвиг влево)

Таблица 8-189 Инструкции ROR и ROL

LAD/FBD	SCL	Описание

	`out := ROL(`
	`  in:=_variant_in_,`
	`  n:=_uint_in);`
	`out := ROR(`
	`  in:=_variant_in_,`
	`  n:=_uint_in);`
		С помощью инструкций циклического сдвига (ROR и ROL) выполняется циклический сдвиг битовой комбинации параметра IN. Результат присваивается параметру OUT. Параметр N устанавливает число позиций битов для циклического сдвига.
		• ROR: Циклический сдвиг битовой комбинации вправо
		• ROL: Циклический сдвиг битовой комбинации влево

[1] В LAD и FBD: Нажать на "???" и выбрать типы данных из выпадающего списка.

Таблица 8-190 Типы данных для параметров

Параметр	Тип данных	Описание
IN	Целые числа	Битовая комбинация для циклического сдвига
N	USInt, UDint	Число позиций битов для циклического сдвига
OUT	Целые числа	Битовая комбинация после циклического сдвига

- При N=0 циклический сдвиг не выполняется. Значение IN присваивается OUT.
- Битовые данные циклически сдвигаются и значения, которые выталкиваются с одной стороны, вставляются с другой.
- Если число позиций битов для циклического сдвига (N) превышает число битов в целевом значении (8 для Byte, 16 для Word, 32 дляDWord), то циклический сдвиг все же выполняется.
- ENO после выполнения инструкции циклического сдвига всегда имеет состояние TRUE.

Таблица 8-191 Пример: ROR с данными Word

Циклический сдвиг битов с правой стороны влево (N = 1)			
IN	0100 0000 0000 0001	Значение OUT до первого циклического сдвига:	0100 0000 0000 0001
		После первого циклического сдвига вправо:	1010 0000 0000 0000
		После второго циклического сдвига вправо:	0101 0000 0000 0000

Расширенные инструкции

9

9.1 Функции даты, времени и часов

9.1.1 Инструкции для работы с датой и временем

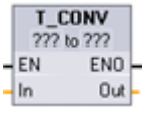
Инструкции для даты и времени используются для расчетов, связанных с календарем и временем.

- T_CONV преобразует значение в (или из) типы(ов) данных даты и типы(ов) данных размерности byte, word и dword.
- T_ADD складывает значения Time и DTL: (Time + Time = Time) или (DTL + Time = DTL)
- T_SUB вычитает значения Time и DTL: (Time - Time = Time) или (DTL - Time = DTL)
- T_DIFF выводит разность между двумя значениями DTL как значение Time: DTL - DTL = Time
- T_COMBINE объединяет значения Date и Time_and_Date для получения значения DTL

Информацию о формате данных DTL и Time можно найти в разделе Типы данных времени и даты (Страница 127).

Таблица 9-1 Инструкция T_CONV (преобразование и извлечение значений времени)

LAD/FBD	Пример SCL	Описание

	`out := DINT_TO_TIME( in:=_variant_in);`
	
	`out := TIME_TO_DINT( in:=_variant_in);`
		T_CONV преобразует значение в (или из) типы(ов) данных даты и типы(ов) данных размерности byte, word и dword.

[1] В LAD и FBD: Нажать на "???" и выбрать типы данных для источника и цели из выпадающего списка.

[2] В SCL: Перетащить инструкцию T_CONV их каталога инструкций в редактор текстов программ. После выбрать типы данных для источника и цели.

Таблица 9-2 Действительные типы данных для преобразованийT_CONV

Тип данных IN (или OUT)	Типы данных OUT (или IN)
TIME (миллисекунды)	DInt, Int, SInt, UDInt, UInt, USInt, TOD Только в SCL: Byte, Word, Dword
DATE (число дней, начиная с 1 января 1990 года)	DInt, Int, SInt, UDInt, UInt, USInt, DTL Только в SCL: Byte, Word, Dword
TOD (число миллисекунд, начиная с полуночи - 24:00:00 000)	DInt, Int, SInt, UDInt, UInt, USInt, TIME, DTL Только в SCL: Byte, Word, Dword

Примечание

Использование T_CONV для преобразования больших объемов данных в меньшие объемы данных

При преобразовании типа данных, содержащего большое количество байт, в тип данных с меньшим размером, значения данных могут быть обрезаны. При возникновении такой ошибки ENO сбрасывается на 0.

Преобразование в(из) тип(а) данных DTL

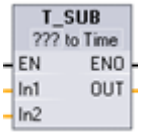
DTL (Date and Time Long) содержит информацию о годе, месяце, дате и времени. Данные DTL могут быть преобразованы в(из) тип(а) данных DATE и TOD.
DTL-преобразование только типа данных "DATE" оказывает влияние только на значения года, месяца и дня. DTL-преобразование с "TOD"-данными оказывает влияние только на значения часов, минут и секунд.

При преобразовании T_CONV в DTL, незадействованные элементы данных в DTL-формате остаются без изменений.

Таблица 9-3 Инструкции T_ADD (суммирование времени) und T_SUB (вычитание времени)

LAD/FBD	SCL	Описание

	`out := T_ADD(`
	`  in1:=_variant_in,`
	`  in2:=_time_in);`
		T_ADD прибавляет значение на входе IN1 (тип данных DTL или Time) к значению на входе IN2 (тип данных) Time). Параметр OUT выводит результат как значение с типом данных DTL или Time. Возможны две инструкции типов данных:
		• Time + Time = Time
		• DTL + Time = DTL

	`out := T_SUB(`
	`  in1:=_variant_in,`
	`  in2:=_time_in);`
		T_SUB вычитает значение на входе IN2 (тип данных Time) из значения IN1 (тип данных DTL или Time). Параметр OUT выводит разностное значение как тип данных DTL или Time. Возможны две инструкции типов данных:
		• Time - Time = Time
		• DTL - Time = DTL

[1] В LAD и FBD: Нажать на "???" и выбрать типы данных из выпадающего списка.

Таблица 9-4 Типы данных для параметров T_ADD и T_SUB

Параметр и тип данных		Тип данных	Описание
IN1[1]	IN	DTL, Time	Значение DTL или Time
IN2	IN	Time	Прибавляемое или вычитаемое значениеTime
OUT	OUT	DTL, Time	Сумма или разность DTLили Time

[1] Выбрать тип данных для IN1 из выпадающего списка под именем инструкции. При выборе типа данных для IN1 устанавливается и тип данных для параметра OUT.

Таблица 9-5 Оператор T_DIFF (разность времен)

LAD/FBD	SCL	Описание

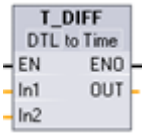
	`out := T_DIFF(`
	`in1:=_DTL_in,`
	`in2:=_DTL_in);`
		T_DIFF вычитает значение DTL (IN2) из значенияDTL (IN1). Параметр OUT выводит разностное значение как тип данных Time.
		• DTL - DTL = Time

Таблица 9-6 Типы данных для параметров T_DIFF

Параметр и тип данных		Тип данных	Описание
IN1	IN	DTL	Значение DTL
IN2	IN	DTL	Вычитаемое значение DTL
OUT	OUT	Time	Разность Time

Коды условий: ENO = 1 означает, что ошибок не обнаружено. ENO = 0 и параметр OUT = 0 - ошибки:

- Недопустимое значение DTL
- Недопустимое значение Time

Таблица 9-7 Инструкция T_COMBINE (объединение времени)

LAD/FBD	SCL	Описание

	`out :=`
	`CONCAT_DATE_TOD(`
	`In1 := _date_in,`
	`In2 := _tod_in);`
		T_COMBINE объединяет значения Date и Time_of_Day для получения значения DTL.

[1] Следует помнить, что инструкция T_COMBINE соответствует в расширенных инструкциях функции CONCAT_DATE_TOD в SCL.

Таблица 9-8 Типы данных для параметров T_COMBINE

Параметр и тип данных		Тип данных	Описание
IN1	IN	Date	Значение Date, используемое для объединения, должно находиться в пределах между DATE#1990-01-01 и DATE#2089-12-31
IN2	IN	Time_of_Day	Значения Time_of_Day, используемые для объединения
OUT	OUT	DTL	Значение DTL

9.1.2 Функции часов

⚠ ПРЕДУПРЕЖДЕНИЕ

Опасность несанкционированного доступа к сетям пользователя при синхронизации по протоколу NTP (Network Time Protocol)

Если злоумышленник получит доступ к сетям пользователя посредством синхронизации по протоколу сетевого времени (NTP), то он может начать управлять процессом, изменив системное время CPU. Такая передача контроля над процессом может привести к смерти, серьезным травмам или материальному ущербу.

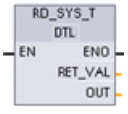
Функция клиента NTP для S7-1200 CPU по умолчанию отключена, и, после ее включения, только сконфигурированные IP-адреса могут выступать в качестве сервера NTP. CPU отключает эту функцию по умолчанию, и необходимо настроить эту функцию, чтобы разрешить удаленную коррекцию системного времени CPU.

S7-1200 CPU поддерживает прерывания и операции по времени, которые зависят от точности системного времени CPU. Если настраивается NTP и разрешается синхронизацию времени с сервера, то следует убедиться, что сервер является доверенным источником. Невыполнение этого требования может привести к нарушениям в системе безопасности, которые могут позволить неизвестному пользователю взять на себя управление процессом, переставив системное время CPU.

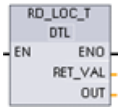
Для получения информации и рекомендаций по обеспечению безопасности см. Operational Guidelines for Industrial Security (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) на сайте поддержки Siemens.

С помощью clock-инструкций можно устанавливать и считывать системные часы CPU. Тип данных DTL (Страница 127) предоставляет значения для даты и времени.

Таблица 9-9 Инструкции для системного времени

LAD/FBD	SCL	Описание
	`ret_val := WR_SYS_T( in:=_DTL_in_);`	WR_SYS_T (установка реального времени) устанавливает часы реального времени в CPU на основе значения DTL из параметра IN. Это значение времени не учитывает ни локальный часовой пояс, ни перестановку часов при переходе на "замнее/летнее" время.
	`ret_val := RD_SYS_T( out=>_DTL_out);`	С помощью RD_SYS_T (считывание реального времени) считывается текущее системное время CPU. Это значение времени не учитывает ни локальный часовой пояс, ни перестановку часов при переходе на "замнее/летнее" время.

LAD/FBD	SCL	Описание

	`ret_val :=`
	`RD_LOC_T(`
	
	`out=>_DTL_out);`
		RD_LOC_T (считывание местного времени) предоставляет текущее местное время CPU как тип данных DTL. Это значение времени учитывает местный часовой пояс и переход на зимнее/летнее время (если сконфигурировано).

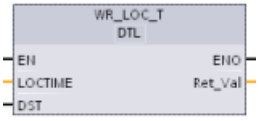
	`ret_val :=`
	`WR_LOC_T(`
	
	`LOCTIME:=DTL_in_,`
	`DST:_in_;`
		WR_LOC_T (установка местного времени) устанавливает дату и время часов CPU. Дата и время назначаются через параметр LOCTIME с типом данных DTL как местное время. С помощью структуры DB "TimeTransformationRule (Страница 348)" инструкция рассчитывает системное время. Степень детализации данных для местного и системного времени зависит от продукта и составляет как минимум одну миллисекунду. Если входные значения на параметре LOCTIME меньше, чем поддерживаемые CPU, то эти значения округляются при расчете системного времени.
		Примечание: Необходимо установить свойства "Реального времени" (Time of day) в конфигурации устройства (часовой пояс, активация летнего времени, начало и конец летнего времени). В ином случае WR_LOC_T не сможет интерпретировать переход на летнее время.

Таблица 9-10 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
IN	IN	DTL	Реальное время для установки в системных часах CPU
OUT	OUT	DTL	RD_SYS_T: Текущее системное время CPU RD_LOC_T: Текущее местное время, с учетом перехода на летнее/зимнее время (если сконфигурировано)
LOCTIME	IN	DTL	WR_LOC_T: Местное время
DST	IN	BOOL	WR_LOC_T: **D**aylight **S**aving **T**ime (летнее время) обрабатывается только в течение "двойного" часа при переходе на летнее время. • TRUE = летнее время (первый час) • FALSE = зимнее время (второй час)
RET_VAL	OUT	Int	Условие выполнения

- Для расчета местного времени используются часовой пояс и настройки для перехода на летнее/зимнее время, введенные на вкладке "Общее" конфигурации устройства в разделе "Реальное время".
- Часовой пояс рассчитывается по системному времени UTC или GMT.
- Для перехода на летнее время необходимо ввести месяц, неделю, день и час перевода часов на летнее время.

- Для перехода на зимнее время также необходимо ввести месяц, неделю, день и час перевода часов на зимнее время.
- Смещение часового пояса всегда относится к значению системного времени. Смещение летнего времени применяется только в том случае, если выполняется переход на летнее время.

Примечание

Установка перехода на летнее/зимнее время

В свойствах "Реального времени" для "Перехода на летнее/зимнее время" в конфигурации устройства CPU должно быть установлено местное время.

Коды условий: ENO = 1 означает, что ошибок не обнаружено. ENO = 0 означает наличие ошибки выполнения, на выходе RET_VAL может быть считан код условия.

RET_VAL (W#16#....)	Описание
0000	Текущее местное время указывается как зимнее время.
0001	Было сконфигурировано летнее время, и текущее местное время указывается как летнее время.
8080	Местное время недоступно или неправильное значение LOCTIME.
8081	Недопустимое значение года или времени, назначенное параметром LOCTIME, является недействительным
8082	Недопустимое значение для месяца (байт 2 в DTL формате)
8083	Недопустимое значение для дня (байт 3 в DTL формате)
8084	Недопустимое значение для часов (байт 5 в DTL формате)
8085	Недопустимое значение для минут (байт 6 в DTL формате)
8086	Недопустимое значение для секунд (байт 7 в DTL формате)
8087	Недопустимое значение для наносекунд (байты с 8 по 11 в DTL формате)
8089	Значение времени отсутствует (час при переходе на летнее время уже прошел)
80B0	Часы реального времени вышли из строя.
80B1	Структура "TimeTransformationRule" не определена.

9.1.3 Структура данных TimeTransformationRule

Описание

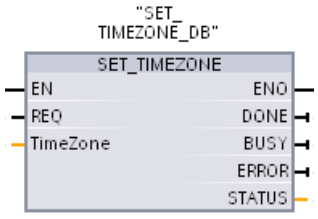
Правила переключения между летним и зимним временем определены в структуре TimeTransformationRule. Структура выглядит следующим образом:

Название		Тип данных	Описание
TimeTransformationRule		STRUCT	
	Bias	INT	Разность между местным временем и UTC [в минутах] Диапазон: от -1439 до 1439
	DaylightBias	INT	Разность между летним временем и зимним временем [в минутах] Диапазон: От 0 до 60

Название		Тип данных	Описание
	DaylightStartMonth	USINT	Месяц перехода на летнее время Диапазон: От 1 до 12
	DaylightStartWeek	USINT	Неделя перехода на летнее время 1 = Первое появление дня недели в месяце, ..., 5 = Последнее появление дня недели в месяце
	DaylightStartWeekday	USINT	День недели перехода на летнее время: 1 = Воскресенье
	DaylightStartHour	USINT	Час перехода на летнее время Диапазон: От 0 до 23
	DaylightStartMinute	USINT	Минута перехода на летнее время Диапазон: От 0 до 59
	StandardStartMonth	USINT	Месяц перехода на зимнее время Диапазон: От 1 до 12
	StandardStartWeek	USINT	Неделя перехода на зимнее время 1 = Первое появление дня недели в месяце, ..., 5 = Последнее появление дня недели в месяце
	StandardStartWeekday	USINT	День недели перехода на зимнее время: 1 = Воскресенье
	StandardStartHour	USINT	Час перехода на зимнее время Диапазон: От 0 до 23
	StandardStartMinute	USINT	Минута перехода на зимнее время Диапазон: От 0 до 59
	TimeZoneName	STRING[80]	Название часового пояса: "(GMT +01:00) Амстердам, Берлин, Берн, Рим, Стокгольм, Вена"

9.1.4 Инструкция SET_TIMEZONE (установка часового пояса)

Таблица 9-11 Инструкция SET_TIMEZONE

LAD/FBD	SCL	Описание
	**"SET_TIMEZONE_DB"(** **REQ:=_bool_in,** **Timezone:=_struct_in,** **DONE=>_bool_out_,** **BUSY=>_bool_out_,** **ERROR=>_bool_out_,** **STATUS=>_word_out_);**	Установка параметров часового пояса и перехода на летнее/зимнее время, которые используются для преобразования системного времени CPU в местное время.

[1] В примере SCL "SET_TIMEZONE_DB" - это имя DB экземпляра.

Таблица 9-12 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	REQ =1: Выполнение функции
Timezone	IN	TimeTransformationRule	Правила для преобразования системного времени в местное время
DONE	OUT	Bool	Функция завершена
BUSY	OUT	Bool	Функция занята
ERROR	OUT	Bool	Обнаружена ошибка
STATUS	OUT	Word	Результат функции / сообщение об ошибке

Для конфигурирования параметров часового пояса для CPU вручную, следует использовать свойства "Реального времени" на вкладке "Общее" конфигурации устройства.

С помощью инструкции SET_TIMEZONE настраивается местное время. Параметры структуры "TimeTransformationRule (Страница 348)" задают часовой пояс и установки для автоматического переключения летнего и зимнего времени.

Примечание

Влияние инструкции SET_TIMEZONE на флэш-память

Инструкция SET_TIMEZONE отвечает за процессы записи во флэш-память (во внутреннюю загружаемую память или на карту памяти). Для увеличения срока службы флэш-памяти следует использовать инструкцию SET_TIMEZONE для сокращения числа обновлений.

Коды условий: ENO = 1 означает, что ошибок не обнаружено. ENO = 0 означает наличие ошибки выполнения, на выходе STATUS может быть считан код условия.

STATUS (W#16#....)	Описание
0	Ошибки отсутствуют
7000	Нет активных заданий на обработку.
7001	Запуск обработки задания. Параметр BUSY = 1, DONE = 0
7002	Промежуточный вызов (REQ не играет роли): Инструкция уже активна; BUSY имеет значение "1".
808x	Ошибка компонента "x": Например, 8084 означает, что DaylightStartWeekif это не значение от 1 до 5.

9.1.5 RTM (счетчик часов работы)

Таблица 9-13 Инструкция RTM

LAD/FBD	SCL	Описание

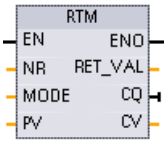
	`RTM(NR:=_uint_in_,`
	`MODE:=_byte_in_,`
	`PV:=_dint_in_,`
	`CQ=>_bool_out_,`
	`CV=>_dint_out_);`
		Инструкция RTM (счетчик часов работы) может устанавливать, запускать, останавливать и считывать информацию счетчиков рабочего времени в CPU.

Таблица 9-14 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
NR	IN	UInt	Номер счетчика часов работы (возможные значения: 0..9)
MODE	IN	Byte	Номер режима выполнения RTM: • 0 = Выборка значений (состояние записывается в CQ, а текущее значение в CV) • 1 = Start (запуск с последнего значения счетчика) • 2 = Stop (Стоп) • 4 = Set (установка на значение, указанное в PV) • 5 = Set (установка на значение, указанное в PV) с последующим запуском • 6 = Set (установка на значение, указанное в PV) с последующим остановом • 7 = Сохранение всех RTM значений CPU на карте памяти (MC)
PV	IN	DInt	Предустановка значение часов для указанного счетчика часов работы
RET_VAL	OUT	Int	Результат функции / сообщение об ошибке
CQ	OUT	Bool	Состояние счетчика часов работы (1 = работает)
CV	OUT	DInt	Текущее значение часов работы для указанного счетчика часов работы

CPU поддерживает до 10 счетчиков часов работы, отслеживая длительность рабочего цикла критических подсистем управления. Каждый отдельный счетчик часов работы должен быть запущен по отдельности с помощью инструкции RTM. Все счетчики часов работы останавливаются при переходе CPU из RUN в STOP. С помощью режима выполнения 2 инструкции RTM можно останавливать отдельные счетчики.

При переходе CPU из STOP в RUN необходимо перезапустить все счетчики часов работы с помощью инструкции RTM для каждого счетчика. При превышении счетчиком часов работы значения 2.147.483.647 часов, процесс счета завершается и появляется сообщение об ошибке переполнения. Для сброса счетчика или изменения его значения необходимо выполнить инструкцию RTM один раз для каждого из счетчика.

При отключении питания или перезапуске CPU текущие значения счетчиков часов работы записываются в сохраняющую память. При запуске CPU сохраненные значения счетчиков часов работы заново загружаются в счетчики, т.е. суммарные значения времени работы не теряются. Для выполнения дальнейшего суммирования рабочего времени счетчики часов работы должны быть перезапущены.

Кроме этого, программа предлагает возможность с помощью режима выполнения 7 инструкцииRTM сохранять значения счетчика часов работы на карту памяти. На момент выполнения режима 7 инструкцииRTM состояния всех счетчиков часов работы сохраняются на карту памяти. Эти сохраненные значения со временем могут потерять свою актуальность из-за старт/стоп процессов счетчиков часов работы. Для регистрации важных рабочих процессов, значения на карте памяти должны регулярно обновляться. Преимущество сохранения значений RTM на карту памяти заключается в том, что карта памяти может быть вставлена в запасной CPU, где будут доступны программа и

сохраненные значения RTM. Если значения RTM не были сохранены на карту памяти, то значения рабочего времени теряются (в случае использования запасного CPU).

Примечание

Избегать слишком большого количества операций записи на карту памяти

Количество операций записи на карту памяти должно быть минимальным, чтобы продлить срок службы карты памяти.

Таблица 9-15 Коды условий

RET_VAL (W#16#....)	Описание
0	Ошибки отсутствуют
8080	Неверный номер счетчика часов работы
8081	Отрицательный значение было передано в параметр PV
8082	Переполнение счетчика часов работы
8091	Входной параметр MODE содержит недопустимое значение.
80B1	Значение не может быть сохранено на MC (MODUS = 7)

9.2 Строчные и символьные инструкции

9.2.1 Тип данных String (строка)

Тип данных STRING

Данные типа String хранятся в виде 2-байтного заголовка, за которым следует до 254 байт символов в коде ASCII. Заголовок для данных типа String содержит два байта для длины. Первый байт содержит максимальную длину строки, которая указывается в квадратных скобках при инициализации строки, или устанавливается на 254 по умолчанию. Второй байт заголовка – это текущая длина, равная числу действительных символов в строке. Текущая длина не должна превышать максимальную длину. Число сохраняемых байтов, занимаемых форматом String, на 2 байта больше, чем максимальная длина.

Инициализация строковых данных

Входные и выходные строковые данные должны быть инициализированы в памяти как действительные строки перед исполнением инструкций.

Действительные строковые данные

Действительные строковые данные имеют максимальную длину, которая должна быть больше нуля, но меньше 255. Текущая длина не должна превышать максимальную длину.

Строковые данные не могут назначаться областям памяти I или Q.

Дополнительную информацию см. в: Формат данных типа STRING (Страница 129).

9.2.2 S_MOV (перемещение строки символов)

Таблица 9-16 Инструкция перемещения строки символов

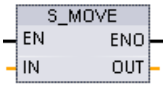
LAD/FBD	SCL	Описание
	`out := in;`	Копирование строки символов источника IN на адрес OUT. Выполнение S_MOVE не влияет на содержание исходной строки символов.

Таблица 9-17 Типы данных для параметров

Параметр	Тип данных	Описание
IN	String	Исходная строка символов
OUT	String	Целевой адрес

Если фактическая длина строки символов на входе N превышает максимальную длину сохраненной на выходе OUT строки символов, то копируется только часть строки IN, соответствующая OUT по длине.

9.2.3 Инструкции для преобразования строк символов

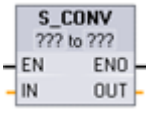
9.2.3.1 Инструкции S_CONV, STRG_VAL и VAL_STRG (преобразование из/в строку символов и числовое значение)

С помощью следующих инструкций можно преобразовывать числовые строки в числовые значения и числовые значения в числовые строки:

- С помощью S_CONV выполняется преобразование числовое строки в числовое значение и числового значения в числовую строку.
- С помощью STRG_VAL выполняется преобразование числовое строки в числовое значение с опциями форматирования.
- С помощью VAL_STRG выполняется преобразование числового значения в числовую строку с опциями форматирования.

S_CONV (преобразование строки символов)

Таблица 9-18 Инструкции для преобразования строк символов

LAD/FBD	SCL	Описание
	`out := <Type>_TO_<Type>(in);`	Преобразует строку символов в соответствующее значение или значение в соответствующую строку символов. У инструкции S_CONV нет опций для форматирования выходного значения. Поэтому инструкция S_CONV является более простой, но менее гибкой, чем инструкции STRG_VAL и VAL_STRG.

[1] В LAD/FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.

[2] В SCL: Выбрать S_CONV aus расширенных инструкций и типы данных для преобразования. После STEP 7 предоставляет соответствующую инструкцию для преобразования.

Таблица 9-19 Типы данных (преобразование строки символов в числовое значение)

Параметр и тип данных		Тип данных	Описание
IN	IN	String, WString	Введенная строка символов
OUT	OUT	String, WString, Char, WChar, SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Выводимое числовое значение

Преобразование параметра строки символов IN начинается с первого символа и продолжается до конца строки символов или до первого символа, не являющегося "0" до "9", "+", "-" или ".". Результат доступен по адресу, указанному с помощью параметра OUT. Если выводимое числовое значение не соответствует диапазону типа данных OUT, то параметр OUT устанавливается на 0, а ENO - на FALSE. В ином случае параметр OUT содержит правильный результат и ENO устанавливается на TRUE.

Правила форматирования для ввода строк символов:

- Если строка символов в IN содержит десятичный знак, то следует использовать символ ".".
- Запятые "," в качестве разделителя тысяч разрядов слева от десятичной точки допускаются, но игнорируются.
- Начальные пробелы игнорируются.

S_CONV (преобразование значения в строку символов)

Таблица 9-20 Типы данных (преобразование числового значения в строку символов)

Параметр и тип данных		Тип данных	Описание
IN	IN	String, WString, Char, WChar, SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Введенное числовое значение
OUT	OUT	String, WString	Выводимая строка символов

Целочисленное значение с или без знака или значение с плавающей запятой IN преобразуется в соответствующую символьную строку, доступную в параметре OUT.

Параметр OUT должен ссылаться на допустимую строку символов до выполнения преобразования. Допустимая строка содержит максимальную длину строки в первом байте, текущую длину строки во втором байте и символы текущей строки в следующих байтах. Преобразованная строка символов заменяет символы в строке символов OUT, начиная с первого символа, и соответствующим образом согласует байт с фактической длиной выведенной строки символов. Байт с максимальной длиной выведенной строки символов не изменяется.

Число заменяемых символов зависит от типа данных и числового значения параметра IN. Замененное число символов должно поместиться в определенную в параметре OUT длину строки символов. Максимальная длина строки символов (первый байт) строки символов в параметре OUT не должна быть меньше максимального числа ожидаемых преобразованных символов. В приведенной ниже таблице представлены примеры преобразования значений в строки символов с помощью S_CONV:

Правила форматирования для вывода строк символов:

- Значения, записываемые в параметр OUT, не используют ведущий знак "+".
- Используется представление чисел с фиксированной точкой (не экспоненциальное представление).
- Если параметр IN имеет тип данных Real, то для разделения целой и дробной части числа используется десятичная точка ".".
- Значения в выводимой строке символов выравниваются по правому краю, им предшествуют пробелы, которые заполняют пустые позиции символов.

Таблица 9-21 Максимальная длина строки символов для каждого типа данных

Тип данных IN	Определяемые S_CONV знакоместа	Пример для преобразованной символьной строки[1]	Общая длина строки символов с байтам для максимальной и фактической длины
USInt	4	"x255"	6
SInt	4	"-128"	6
UInt	6	"x65535"	8
Int	6	"-32768"	8
UDInt	11	"x4294967295"	13
DInt	11	"-2147483648"	13
Real	14	"x-3.402823E+38" "x-1.175495E-38" "x+1.175495E-38" "x+3.402823E+38"	16
LReal	21	"-1.7976931348623E+308" "-2.2250738585072E-308" "+2.2250738585072E-308" "+1.7976931348623E+308"	23

[1] Символы "x" представляют собой пробелы, которыми заполняют пустые позиции при выравнивании по правому краю поля, выделенного для преобразованного значения.

STRG_VAL (преобразование строки символов в числовое значение)

Таблица 9-22 Инструкция для преобразования строки символов в числовое значение

LAD/FBD	SCL	Описание

	`"STRG_VAL"(`
	`in:=_string_in,`
	`format:=_word_in,`
	`p:=uint_in,`
	`out=>_variant_out);`
		Преобразует числовую строку в соответствующие целые числа или числа с плавающей запятой.

[1] В LAD/FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.

Таблица 9-23 Типы данных для инструкции STRG_VAL

Параметр и тип данных		Тип данных	Описание
IN	IN	String, WString	Строка символов ASCII для преобразования
FORMAT	IN	Word	Опции для выходных форматов
P	IN	UInt, Byte, USInt	IN: Ссылка на первый символ, который должен быть преобразован (первый символ = 1)
OUT	OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Преобразованное числовое значение

Преобразование начинается во введенной строке символов на знаковом сдвиге P и продолжается до конца строки символов или до первого символа, не являющегося "+", "-", ".", ",", "e", "E" или одним из символов от "0" до "9". Результат доступен по адресу, указанному в параметре OUT.

Строковые данные должны быть инициализированы в памяти как действительная строка символов перед исполнением инструкций.

Параметр FORMAT инструкции STRG_VAL определен ниже. Свободные позиции битов должны быть установлены на ноль.

Таблица 9-24 Формат инструкции STRG_VAL

Бит 16							**Бит 8**	**Бит 7**							**Бит 0**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	f	r

f = формат представления 1= экспоненциальное представление
0 = представление с фиксированной точкой

r = формат с десятичной точкой (запятая)
0 = "." (точка)

Таблица 9-25 Значения параметра FORMAT

FORMAT (W#16#)	Формат представления	Представление десятичного разделителя
0000 (по умолчанию)	Фиксированная точка	"."
0001		","
0002	Экспоненциальное предста-вление	"."
0003		","
от 0004 до FFFF	Недопустимые значения	

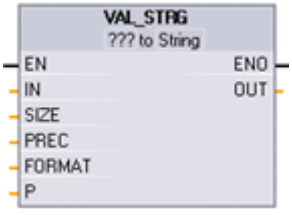
Правила для преобразования STRG_VAL:

- Если в качестве десятичного разделителя используется символ точки ".", то запятые "," слева от десятичной точки интерпретируются как разделители тысяч. Запятые допускаются, но игнорируются.
- Если в качестве десятичного разделителя используется символ запятой "," то точки "." слева от запятой интерпретируются как символы разделения тысяч. Такие точки допускаются, но игнорируются.
- Начальные пробелы игнорируются.

VAL_STRG (преобразование числового значения в строку символов)

Таблица 9-26 Инструкция преобразования числового значения в строку символов

LAD/FBD	SCL	Описание

	`"VAL_STRG"(`
	`in:=_variant_in,`
	`size:=_usint_in,`
	`prec:=_usint_in,`
	`format:=_word_in,`
	`p:=uint_in,`
	`out=>_string_out);`
		Преобразует целочисленное значение с или без знака или значение с плавающей запятой в соответствующее строковое представление.

[1] В LAD/FBD: Нажать на "???" и выбрать тип данных из выпадающего списка.

Таблица 9-27 Типы данных для инструкции VAL_STRG

Параметр и тип данных		Тип данных	Описание
IN	IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Значение для преобразования
SIZE	IN	USInt	Число символов, подлежащих записи в строку символов OUT
PREC	IN	USInt	Точность или размер дробной части. Без десятичного разделителя.
FORMAT	IN	Word	Опции для выходных форматов
P	IN	UInt, Byte, USInt	IN: Указатель на первый символ строки OUT, который должен быть заменен (первый символ = 1)
OUT	OUT	String, WString	Преобразованная строка символов

Эта инструкция преобразует представленное в параметре IN значение в строку символов, адресуемую в параметре OUT. Параметр OUT должен ссылаться на допустимую строку символов до выполнения преобразования.

Преобразованная строка символов заменяет символы строки OUT, начиная с определенного в параметре P знакового сдвига, до установленного параметром SIZE числа символов. Количество символов в SIZE должно соответствовать длине строки символов OUT, начиная с позиции символа P. Если параметр SIZE равен нулю, то символа на позиции P в строке символов OUT заменяются без ограничений. Эта инструкция полезна для встраивания цифровых символов в текстовую символьную строку. Например, можно поместить цифры "120" в символьную строку "Давление насоса = 120 кПа".

Параметр PREC указывает точность или число знаков после запятой в символьной строке. Если параметр IN это целочисленное значение, то PREC указывает на позицию десятичного разделителя. Если, например, значение данных 123 и PREC = 1, то результат будет "12.3". Максимальная поддерживаемая точность для типа данных Real составляет 7 цифр.

Если параметр P больше, чем фактический размер строки символов OUT, то до позиции P вставляются пробелы, а результат прикрепляется в конце символьной строки. Преобразование завершается по достижении максимальной длины символьной строки OUT.

Параметр FORMAT инструкции VAL_STRG определен ниже. Свободные позиции битов должны быть установлены на ноль.

Таблица 9-28 Формат инструкции VAL_STRG

Бит 16							**Бит 8**	**Бит 7**							**Бит 0**
0	0	0	0	0	0	0	0	0	0	0	0	0	s	f	r

s = символ для знака числа — 1= знаки "+" и "-"
0 = только знак "-"

f = формат представления — 1= экспоненциальное представление
0 = представление с фиксированной точкой

r = формат с десятичной точкой — (запятая)
0 = "." (точка)

Таблица 9-29 Значения параметра FORMAT

FORMAT (WORD)	**Знак**	**Формат представления**	**Представление десятичного разделителя**
W#16#0000	Только "-"	Фиксированная точка	"."
W#16#0001			","
W#16#0002		Экспоненциальное представление	"."
W#16#0003			","

FORMAT (WORD)	Знак	Формат представления	Представление десятичного разделителя
W#16#0004	"+" и "-"	Фиксированная точка	"."
W#16#0005			","
W#16#0006		Экспоненциальное представление	"."
W#16#0007			","
От W#16#0008 до W#16#FFFF	Недопустимые значения		

Правила форматирования для строки символов в параметре OUT:

- Если преобразованная строка символов короче указанного размера, то слева к ней добавляются ведущие пробелы.
- Если знаковый бит параметра FORMAT имеет значение FALSE, то целочисленные значения с и без знака записываются в выходной буфер без ведущего знака "+". В случае необходимости используется знак "-".
  <ведущие пробелы><цифры без ведущих нулей>'.'<цифры PREC>
- Если знаковый бит параметра имеет значение TRUE, то целочисленные значения с и без знака всегда записываются в выходной буфер с ведущим знаком.
  <ведущие пробелы><знак><цифры без ведущих нулей>'.'<цифры PREC>
- Если в качестве FORMAT устанавливается экспоненциальное представление, то значения, имеющие тип данных Real, записываются в выходной буфер следующим образом:
  <ведущие пробелы><знак><цифра> '.' <цифры PREC>'E' ><знак><цифры без ведущего нуля>
- Если в качестве параметра FORMAT устанавливается представление с фиксированной точкой, то целочисленные значения с или без знака и вещественные значения записываются в выходной буфер следующим образом:
  <ведущие пробелы><знак><цифры без ведущих нулей>'.'<цифры PREC>
- Ведущие нули слева от десятичной точки (кроме цифры, стоящей непосредственно рядом с десятичной точкой) подавляются.
- Значение справа от десятичной точки округляется, чтобы оно соответствовало числу цифр справа от десятичной точки, заданному параметром PREC.
- Размер выводимой строки символов должен, по крайней мере, на три байта превышать число цифр справа от десятичной точки.
- Значения в выводимой строке выравниваются вправо.

Сигнализируемые ENO условия

Если при выполнении операции преобразования возникает ошибка, то инструкция возвращает следующие результаты:

- ENO устанавливается на 0
- OUT устанавливается на 0, или на значение, как показано в примере для преобразования строки символов в значение.
- OUT остается без изменений или на показанном в примерах значении, если OUT это строка символов.

Таблица 9-30 Состояние ENO

ENO	Описание
1	Ошибки отсутствуют
0	Недопустимый или недействительный параметр, например, обращение к несуществующему DB
0	Недопустимая строка с максимальной длиной 0 или 255
0	Недопустимая строка, в которой длина больше, чем максимальная
0	Преобразованное числовое значение слишком велико для указанного типа данных OUT.
0	Максимальный размер строки для параметра OUT должен быть достаточным для числа символов, указанных в параметре SIZE, начиная с позиции символа в параметре P.
0	Недопустимое значение P, где P=0 или P больше, чем текущая длина строки
0	Параметр SIZE должен быть больше параметра PREC.

Таблица 9-31 Пример преобразования строк символов в значения с помощью S_CONV

Строка символов IN	Тип данных OUT	Значение OUT	ENO
"123"	Int или DInt	123	TRUE
"-00456"	Int или DInt	-456	TRUE
"123,45"	Int или DInt	123	TRUE
"+2345"	Int или DInt	2345	TRUE
"00123AB"	Int или DInt	123	TRUE
"123"	Real	123,0	TRUE
"123,45"	Real	123,45	TRUE
"1.23e-4"	Real	1,23	TRUE
"1.23E-4"	Real	1,23	TRUE
"12.345,67"	Real	12345,67	TRUE
"3.4e39"	Real	3,4	TRUE
"-3.4e39"	Real	-3,4	TRUE
"1.17549e-38"	Real	1,17549	TRUE
"12345"	SInt	0	FALSE
"A123"	-/-	0	FALSE
""	-/-	0	FALSE
"++123"	-/-	0	FALSE
"+-123"	-/-	0	FALSE

Таблица 9-32 Примеры преобразования значений в строки с помощью S_CONV

Тип данных	Значение IN	Строка символов OUT[1]	ENO
UInt	123	"xxx123"	TRUE
UInt	0	"xxxxx0"	TRUE
UDInt	12345678	"xxx12345678"	TRUE
Real	+9123,456	"xx+9.123456E+3"	TRUE

Тип данных	Значение IN	Строка символов OUT[1]	ENO
LReal	+9123,4567890123	"xx +9.1234567890123E +3"	TRUE
Real	-INF	"xxxxxxxxxxINF"	FALSE
Real	+INF	"xxxxxxxxxxINF"	FALSE
Real	NaN	"xxxxxxxxxxNaN"	FALSE

[1] Символы "x" представляют собой пробелы, которыми заполняют пустые позиции при выравнивании по правому краю поля, выделенного для преобразованного значения.

Таблица 9-33 Пример: Преобразование STRG_VAL

Строка символов IN	FORMAT (W#16#....)	Тип данных OUT	Значение OUT	ENO
"123"	0000	Int или DInt	123	TRUE
"-00456"	0000	Int или DInt	-456	TRUE
"123,45"	0000	Int или DInt	123	TRUE
"+2345"	0000	Int или DInt	2345	TRUE
"00123AB"	0000	Int или DInt	123	TRUE
"123"	0000	Real	123,0	TRUE
"-00456"	0001	Real	-456,0	TRUE
"+00456"	0001	Real	456,0	TRUE
"123,45"	0000	Real	123,45	TRUE
"123,45"	0001	Real	12345,0	TRUE
"123,45"	0000	Real	12345,0	TRUE
"123,45"	0001	Real	123,45	TRUE
".00123AB"	0001	Real	123,0	TRUE
"1.23e-4"	0000	Real	1,23	TRUE
"1.23E-4"	0000	Real	1,23	TRUE
"1.23E-4"	0002	Real	1.23E-4	TRUE
"12.345,67"	0000	Real	12345,67	TRUE
"12.345,67"	0001	Real	12,345	TRUE
"3.4e39"	0002	Real	+INF	TRUE
"-3.4e39"	0002	Real	-INF	TRUE
"1.1754943e-38" (и меньше)	0002	Real	0,0	TRUE
"12345"	-/-	SInt	0	FALSE
"A123"	-/-	-/-	0	FALSE
""	-/-	-/-	0	FALSE
"++123"	-/-	-/-	0	FALSE
"+-123"	-/-	-/-	0	FALSE

Следующие примеры для преобразований VAL_STRG основаны на строке OUT, инициализированной следующим образом:

"Current Temp = xxxxxxxxxx C"
где символ "x" представляет собой пробелы, назначенные преобразованному значению.

Таблица 9-34 Пример: Преобразование VAL_STRG

Тип данных	Значение IN	P	SIZE	FORMAT (W#16#....)	PREC	Строка символов OUT	ENO
UInt	123	16	10	0000	0	Current Temp = xxxxxxx123 C	TRUE
UInt	0	16	10	0000	2	Current Temp = xxxxxx0.00 C	TRUE
UDInt	12345678	16	10	0000	3	Current Temp = x12345.678 C	TRUE
UDInt	12345678	16	10	0001	3	Current Temp = x12345,678 C	TRUE
Int	123	16	10	0004	0	Current Temp = xxxxxx+123 C	TRUE
Int	-123	16	10	0004	0	Current Temp = xxxxxx-123 C	TRUE
Real	-0,00123	16	10	0004	4	Current Temp = xxx-0.0012 C	TRUE
Real	-0,00123	16	10	0006	4	Current Temp = -1.2300E-3 C	TRUE
Real	-INF	16	10	-/-	4	Current Temp = xxxxxx-INF C	FALSE
Real	+INF	16	10	-/-	4	Current Temp = xxxxxx+INF C	FALSE
Real	NaN	16	10	-/-	4	Current Temp = xxxxxxxNaN C	FALSE
UDInt	12345678	16	6	-/-	3	Current Temp = xxxxxxxxxx C	FALSE

9.2.3.2 Инструкции Strg_TO_Chars и Chars_TO_Strg (преобразование в(из) строку символов и массив из CHAR)

Strg_TO_Chars копирует строку символов ASCII в массив из символьных байтов.

Chars_TO_Strg копирует массив из символьных байтов ASCII в строку символов.

Примечание

Только отсчитываемые от нуля типы массивов (Array [0..n] of Char) или (Array [0..n] of Byte) могут использоваться в качестве входных параметров Chars для инструкции Chars_TO_Strg , или как параметры IN_OUT Chars для инструкции Strg_TO_Chars .

Таблица 9-35 Инструкция Strg_TO_Chars

LAD/FBD	SCL	Описание

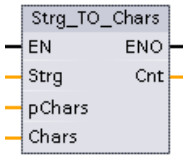
	`Strg_TO_Chars(`
	`  Strg:=_string_in_,`
	`  pChars:=_dint_in_,`
	`  Cnt=>_uint_out_,`
	
	`Chars:=_variant_inout_);`
		Вся входная строка символов Strg копируется в массив из символов в параметре IN_OUTChars..
		Инструкция перезаписывает байты, начиная с номера элемента массива, заданного параметромpChars.
		Могут быть использованы все строки с максимальной поддерживаемой длиной (1 .. 254).
		Конечный ограничитель не прописывается и должен при необходимости устанавливаться пользователем. Для установки конечного ограничителя сразу после последнего записанного символа массива, следует использовать следующий номер элемент массива [pChars+Cnt].

Таблица 9-36 Типы данных для параметров (Strg_TO_Chars)

Параметр и тип данных		Тип данных	Описание
Strg	IN	String, WString	Исходная строка
pChars	IN	DInt	Номер элемента массива для первого строкового символа, записываемого в целевой массив
Chars	IN_OUT	Вариант	Параметр Chars это указатель на отсчитываемый от нуля массив [0..n] из символов, копируемый из входной строки символов. Массив может быть описан в DB или в виде локальных переменных интерфейса блока. Пример: "DB1".MyArray ссылается на значения элемента MyArray [0..10] of Char в DB1.
Cnt	OUT	UInt	Количество скопированных символов

Таблица 9-37 Инструкция Chars_TO_Strg

LAD/FBD	SCL	Описание

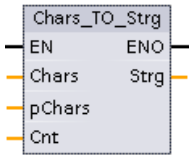
	`Chars_TO_Strg(`
	`  Chars:=_variant_in_,`
	`  pChars:=_dint_in_,`
	`  Cnt:=_uint_in_,`
	`  Strg=>_string_out_);`
		Весь массив или часть массива из символов копируется в строку символов.
		Выходная строка символов должна быть объявлена до выполнения Chars_TO_Strg. После строка символов переписывается инструкцией Chars_TO_Strg.
		Могут быть использованы все строки с максимальной поддерживаемой длиной (1 ... 254).
		Значение для максимальной длины строки символов не изменяется инструкцией Chars_TO_Strg . Копирование из массива и строку символов завершается по достижении максимальной длины символьной строки.
		Символ nul "$00" или значение 16#00 а массиве из символов является разделительным знаком и завершает копирование символов в строку.

Таблица 9-38 Типы данных для параметров (Chars_TO_Strg)

Параметр и тип данных		Тип данных	Описание
Chars	IN	Вариант	Параметр Chars это указатель на отсчитываемый от нуля массив [0..n] из символов, который должен быть преобразован в строку символов. Массив может быть описан в DB или в виде локальных переменных интерфейса блока. Пример: "DB1".MyArray ссылается на значения элемента MyArray [0..10] of Char в DB1.
pChars	IN	Dint	Номер элемента первого копируемого символа в массиве. Элемент массива [0] это значение по умолчанию.
Cnt	IN	UInt	Количество копируемых символов: 0 означает "все"
Strg	OUT	String, WString	Целевая строка символов

Таблица 9-39 Состояние ENO

ENO	Описание
1	Ошибки отсутствуют
0	Chars_TO_Strg: Попытка скопировать в выходную строку больше символьных байтов, чем допускается максимальной длиной описываемой строки
0	Chars_TO_Strg: Значение нулевого символа (16#00) было обнаружено в байтовом массиве входного символа.
0	Strg_TO_Chars: Попытка скопировать в выходной массив больше символьных байтов, чем допускается соответствующим предельным значением для числа элементов

9.2.3.3 ATH и HTA (преобразование в/из строку символов ASCII и шестнадцатеричное число)

Инструкции ATH (преобразование строки ASCII в шестнадцатеричное число) и HTA (преобразование шестнадцатеричного числа в строку ASCII) используются для преобразования между символами ASCII (от 0 до 9 и прописные буквы от A до F) и соответствующим 4-битным шестнадцатеричным представлением.

Таблица 9-40 Инструкция ATH

LAD/FBD	SCL	Описание

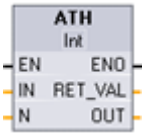
	`ret_val := ATH(`
	`  in:=_variant_in_,`
	`  n:=_int_in_,`
	`  out=>_variant_out_);`
		Преобразование символов ASCII в сжатые шестнадцатеричные числа.

Таблица 9-41 Типы данных для инструкции ATH

Тип параметра		Тип данных	Описание
IN	IN	Вариант	Указатель на байтовый массив символов ASCII
N	IN	UInt	Количество байтовых символов ASCII для преобразования

Тип параметра		Тип данных	Описание
RET_VAL	OUT	Word	Условие выполнения
OUT	OUT	Вариант	Указатель на преобразованный шестнадцатеричный байтовый массив

Преобразование начинается на указанном параметром IN месте и включает в себя N байт. Результат доступен по адресу, указанному в параметре OUT. Можно преобразовать только действительные символы ASCII от 0 до 9, строчные буквы от a до f и прописные буквы от A до F. Любой другой символ будет преобразован в ноль.

8-битные ASCII-кодированные символы преобразуются в 4-битные шестнадцатеричные символы. Два символа ASCII можно преобразовать в один байт, содержащий два 4-битных шестнадцатеричных символа.

Параметры IN и OUT определяют байтовые массивы, а не шестнадцатеричные строковые данные String. Символы ASCII преобразуются и помещаются в шестнадцатеричные символы в том же порядке, в котором они читаются. Если количество ASCII символов нечетное, то в крайний правый полубайт последней преобразованной позиции шестнадцатеричного числа записываются нули.

Таблица 9-42 Примеры: Преобразование строки символов ASCII в шестнадцатеричные числа (ATH)

Символьные байты IN	N	Значение OUT	ENO
'0a23'	4	W#16#0A23	TRUE
'123AFx1a23'	10	16#123AF01023	FALSE
'a23'	3	W#16#A230	TRUE

Таблица 9-43 Инструкция HTA

LAD/FBD	SCL	Описание

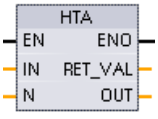
	`ret_val := HTA(`
	`  in:=_variant_in_,`
	`  n:=_uint_in_,`
	`  out=>_variant_out_);`
		Преобразование сжатых шестнадцатеричных чисел в соответствующие символьные ASCII-байты.

Таблица 9-44 Типы данных для инструкции HTA

Параметр и тип данных		Тип данных	Описание
IN	IN	Вариант	Указатель на входной байтовый массив
N	IN	UInt	Количество байтов для преобразования (каждый входной байт содержит два 4-битных полубайта и создает 2N ASCII символа)
RET_VAL	OUT	Word	Условие выполнения
OUT	OUT	Вариант	Указатель на байтовый массив символов ASCII

Преобразование начинается на указанном параметром IN месте и включает в себя N байт. Каждый 4-битный полубайт преобразуется в отдельный 8-битный ASCII-символ и создает на выходе 2N ASCII символьных байт. Все 2N байты выхода записываются в виде ASCII-символов от 0 до 9 или прописными буквами от A до F. Параметр OUT указывает байтовый массив, а не строку.

Каждый полубайт шестнадцатеричного байта преобразуется в символ в том же порядке, в котором они считываются (левый полубайт шестнадцатеричного числа преобразуется первым, за которым следует правый полубайт того же байта).

Таблица 9-45 Примеры: Преобразование шестнадцатеричных чисел в строки символов ASCII (HTA)

Значение IN	N	Символьные байты OUT	ENO (ENO всегда TRUE после выполнения HTA)
W#16#0123	2	'0123'	TRUE
DW#16#123AF012	4	'123AF012'	TRUE

Таблица 9-46 Коды условий ATH and HTA

RET_VAL (W#16#....)	Описание	ENO
0000	Ошибки отсутствуют	TRUE
0007	Недействительный входной ATH символ: Был обнаружен символ, не являющийся ASCII символом 0-9, строчной буквой a-f или прописной буквой A-F	FALSE
8101	Недопустимый или недействительный входной указатель, например, обращение к несуществующему DB.	FALSE
8120	Неправильный формат входной строки символов, т.е. Max. = 0, Max. = 255, текущая строка > Max. или длина в указателе < Max.	FALSE
8182	Входной буфер слишком мал для N	FALSE
8151	Неправильный тип данных для входного буфера	FALSE
8301	Недопустимый или недействительный выходной указатель, например, обращение к несуществующему DB.	FALSE
8320	Неправильный формат выходной строки символов, т.е. Max. = 0, Max. = 255, текущая строка > Max. или длина в указателе < Max.	FALSE
8382	Выходной буфер слишком мал для N	FALSE
8351	Неправильный тип данных для выходного буфера	FALSE

9.2.4 Строковые инструкции

Управляющая программа может использовать следующие строковые и символьные инструкции для формирования сообщений оператору и ведения журнала процесса.

9.2.4.1 MAX_LEN (максимальная длина строки символов)

Таблица 9-47 Инструкция для максимальной длины

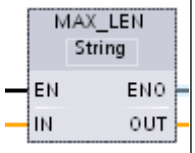
LAD/FBD	SCL	Описание
	`out := MAX_LEN(in);`	MAX_LEN (максимальная длина строки символов) указывает назначенную строке символов IN на выходе OUT максимальную длину. Если во время обработки инструкции возникает ошибка, то строка на выходе будет пустой (нулевой длины). У типов данных String и WString есть две длины: Первый байт (или слово) указывает максимальную длину, а второй байт (или слово) - фактическую длину (текущее количество действительных символов). • Максимальная длина цепочки символов указывается для каждого String или WString описания в квадратных скобках. Число занимаемых строкой String байт на 2 байта больше максимальной длины. Число занимаемых строкой WString слов на 2 слова больше максимальной длины. • Текущая длина отражает количество фактически используемых символов. Текущая длина не должна превышать максимальную длину. Текущая длина указывается в байтах для String и в словах для WString. С помощью инструкции MAX_LEN i можно считать максимальную длину строки символов, а с помощью инструкции LEN - текущую длину строки символов.

Таблица 9-48 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
IN	IN	String, WString	Введенная строка символов
OUT	OUT	DInt	Максимально допустимое число символов в строке символов IN

9.2.4.2 LEN (определение длины строки символов)

Таблица 9-49 Инструкция LEN

LAD/FBD	SCL	Описание
	`out := LEN(in);`	LEN (длина) выводит текущую длину строки символов IN на выходе OUT. Пустая строка символов имеет нулевую длину.

Таблица 9-50 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
IN	IN	String, WString	Введенная строка символов
OUT	OUT	Int, DInt, Real, LReal	Количество действительных символов в строке символов IN

Таблица 9-51 Состояние ENO

ENO	Условие	OUT
1	Недопустимые условия для строки отсутствуют	Правильная длина строки символов
0	Текущая длина IN превышает максимальную длину IN	Текущая длина устанавливается на 0
	Максимальная длина IN не соответствует назначенной области памяти	
	Максимальная длина IN равна 255 (недопустимая длина)	

9.2.4.3 CONCAT (объединение строк символов)

Таблица 9-52 Инструкция для объединения строк символов

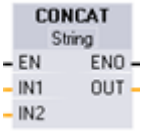
LAD/FBD	SCL	Описание
	`out := CONCAT(in1, in2);`	Инструкция CONCAT (объединение строк символов) соединяет строковые параметры IN1 иIN2 в одну символьную строку, доступную на выходе OUT. После объединения строка IN1 является левой частью, а строка IN2 - правой частью объединенной строки.

Таблица 9-53 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
IN1	IN	String, WString	Введенная строка символов 1
IN2	IN	String, WString	Введенная строка символов 2
OUT	OUT	String, WString	Объединенная строка символов (строка символов 1 + строка символов 2)

Таблица 9-54 Состояние ENO

ENO	Условие	OUT
1	Ошибки отсутствуют	Действительные символы
0	Результирующая строка после объединения больше максимально допустимой длины строки OUT	Символы результирующей строки копируются, пока не будет достигнута максимальная длина строки OUT
	Фактическая длина строки IN1 превышает максимально допустимую длину для строки IN1, фактическая длина строки IN2 превышает максимальную длину для строки IN2, или фактическая длина строки OUT превышает максимальную длину для строки OUT (недействительная строка)	Текущая длина устанавливается на 0
	Максимальная длина IN1, IN2 или OUT не соответствует назначенной области памяти	
	Максимальная длина строк IN1 или IN2 равна 255, максимальная длина строки OUT равна 0 или 255 (тип данных String)	
	Максимальная длина строк IN1 или IN2 равна 65534, максимальная длина строки OUT равна 0 или 65534 (тип данных WString)	

9.2.4.4 Инструкции LEFT, RIGHT и MID (чтение подстрок в строке символов)

Таблица 9-55 Инструкции LEFT, RIGHT и MID

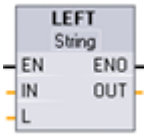
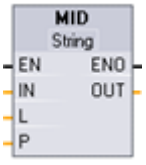
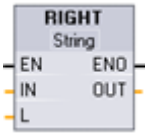
LAD/FBD	SCL	Описание
	`out := LEFT(in, L);`	Инструкция LEFT (левая подстрока) выводит часть строки, состоящую из первых L символов строки параметра IN. • Если L больше, чем текущая длина строки IN, то вся строка IN выводится на выход OUT. • Если вводится пустая строка, то на OUT тоже выводится пустая строка.
	`out := MID(in, L, p);`	Инструкция MID (средняя подстрока) выводит среднюю часть строки. Средняя подстрока, содержащая L символов, начинается с позиции символа P (включительно). Если сумма L и P превышает фактическую длину строкового параметра IN, то выводится часть строки от позиции символа P до конца строки IN.
	`out := RIGHT(in, L);`	Инструкция RIGHT (правая подстрока) выводит последние L символы строки символов. • Если L больше, чем текущая длина строки IN, то вся строка IN выводится на выход OUT. • Если вводится пустая строка, то на OUT тоже выводится пустая строка.

Таблица 9-56 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
IN	IN	String, WString	Введенная строка символов
L	IN	Int	Длина создаваемой подстроки: • LEFT использует часть символов с левого края символьной строки • RIGHT использует часть символов с правого края символьной строки • MID использует часть символов с от позиции P символьной строки
P	IN	Int	Только MID: Позиция первого символа в подстроке, который должен быть скопирован P = 1, для позиции первого символа строки IN
OUT	OUT	String, WString	Выводимая строка символов

Таблица 9-57 Состояние ENO

ENO	Условие	OUT
1	Ошибки отсутствуют	Действительные символы
0	• L или P меньше или равны нулю • P больше максимальной длины строки IN • Фактическая длина IN строки превышает максимальную длину IN, или фактическая длина OUT превышает максимальную длину OUT • Максимальная длина IN, или OUT не соответствует выделенной памяти • Максимальная длина IN или OUT равна 0 или 255 (тип данных String) или 0 или 65534 (тип данных WString)	Текущая длина устанавливается на 0
	Длина (L) подстрок, которая должна быть скопирована, больше максимальной длины строки OUT	Символы копируются, пока не будет достигнута максимальная длина строки OUT.
	Только MID: L или P меньше или равны 0	Текущая длина устанавливается на 0
	Только MID: P больше максимальной длины IN	
	Фактическая длина IN1 превышает максимальную длину IN1 или фактическая длина IN2 превышает максимальную длину IN2 (неправильная строка)	Текущая длина устанавливается на 0
	Максимальная длина IN1, IN2 или OUT не соответствует назначенной области памяти	
	Максимальная длина IN1, IN2 или OUT является недопустимой: 0 или 255 (тип данных String) или 0 или 65534 (тип данных WString)	

9.2.4.5 Инструкция DELETE (удаление символов из строки символов)

Таблица 9-58 Оператор удаления части строки

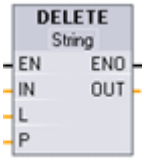
LAD/FBD	SCL	Описание
	`out := DELETE(in, L, p);`	Удаляет L символов из строки IN. Удаление начинается с позиции символа P (включительно), а остальная часть строки выводится в параметре OUT. • Если L равно нулю, то введенная строка выводится на OUT. • Если сумма из L и P больше длины введенной строки, то строка символов удаляется до конца.

Таблица 9-59 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
IN	IN	String, WString	Введенная строка символов
L	IN	Int	Число подлежащих удалению символов
P	IN	Int	Позиция первого символа для удаления: Первый символ строки IN имеет позицию 1.
OUT	OUT	String, WString	Выводимая строка символов

Таблица 9-60 Состояние ENO

ENO	Условие	OUT
1	Ошибки отсутствуют	Действительные символы
0	P больше текущей длины IN	IN копируется в OUT, символы не удаляются
	Результирующая строка после удаления больше максимально допустимой длины строки OUT	Символы результирующей строки копируются, пока не будет достигнута максимальная длина строки OUT
	L меньше 0 или P меньше или равен 0	Текущая длина устанавливается на 0
	Фактическая длина IN превышает максимальную длину IN или фактическая длина OUT превышает максимальную длину OUT	
	Максимальная длина IN, или OUT не соответствует выделенной памяти	
	Максимальная длина IN или OUT равна 0 или 255	

9.2.4.6 INSERT (вставка символов в строку символов)

Таблица 9-61 Инструкция вставка подстроки в строке символов

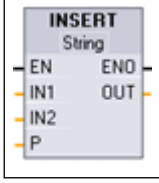
LAD/FBD	SCL	Описание
	`out := INSERT(in1, in2, p);`	Вставляет строку символов IN2 в строку символов IN1. Вставка начинается после символа на позиции P.

Таблица 9-62 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
IN1	IN	String, WString	Введенная строка символов 1
IN2	IN	String, WString	Введенная строка символов 2
P	IN	Int	Последняя позиция символа в строке символов IN1 перед местом вставки для строки символов IN2. Первый символ строки IN1 имеет позицию 1.
OUT	OUT	String, WString	Полученная цепочка символов

Таблица 9-63 Состояние ENO

ENO	Условие	OUT
1	Ошибки отсутствуют	Действительные символы
0	P больше длины IN1	IN2 сразу после последнего символа IN1 объединяется с IN1
	P меньше 0	Текущая длина устанавливается на 0
	Результирующая строка после вставки больше максимально допустимой длины строки OUT	Символы результирующей строки копируются, пока не будет достигнута максимальная длина строки OUT
	Фактическая длина строки IN1 превышает максимально допустимую длину для строки IN1, фактическая длина строки IN2 превышает максимальную длину IN2, или фактическая длина строки OUT превышает максимальную длину для строки OUT (недействительная строка)	Текущая длина устанавливается на 0
	Максимальная длина IN1, IN2 или OUT не соответствует назначенной области памяти	
	Максимальная длина строк IN1 или IN2 равна 255 или максимальная длина строки OUT равна 0 или 255 (тип данных String)	
	Максимальная длина строк IN1 или IN2 равна 65534 или максимальная длина строки OUT равна 0 или 65534 (тип данных WString)	

9.2.4.7 REPLACE (замена символов в строке символов)

Таблица 9-64 Инструкция замены подстроки

LAD/FBD	SCL	Описание

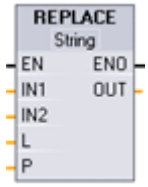
	`out := REPLACE(`
	`in1:=_string_in_,`
	`in2:=_string_in_,`
	`L:=_int_in_,`
	`p:=_int_in);`
		Заменяет L символов в строке символов в параметре IN1. Замена начинается на позиции символа P (включительно) строки IN1, заменяющие символы берутся из параметра IN2.

Таблица 9-65 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
IN1	IN	String, WString	Введенная строка символов
IN2	IN	String, WString	Строка с заменяющими символами
L	IN	Int	Число символов, подлежащих замене
P	IN	Int	Позиция первого символа, подлежащего замене
OUT	OUT	String, WString	Полученная цепочка символов

Если параметр L равен нулю, то строка символов IN2 вставляется на позиции P строки символов IN1, не удаляя символов из строки символов IN1.

Если P равно 1, то первые L символов строки IN1 заменяются символами строки IN2.

Таблица 9-66 Состояние ENO

ENO	Условие	OUT
1	Ошибки отсутствуют	Действительные символы
0	P больше длины IN1	IN2 сразу после последнего символа IN1 объединяется с IN1
	P находится в пределах IN1, но меньше L символов, остающихся в строке IN1	IN2 заменяет конечные символы IN1, начиная с позиции P
	Результирующая строка после замены больше максимально допустимой длины строки OUT	Символы результирующей строки копируются, пока не будет достигнута максимальная длина строки OUT
	Максимальная длина IN1 равна 0	Символы IN2 копируются в OUT
	L меньше 0 или P меньше или равен 0	Текущая длина устанавливается на 0
	Фактическая длина строки IN1 превышает максимально допустимую длину для строки IN1, фактическая длина строки IN2 превышает максимальную длину для строки IN2, или фактическая длина строки OUT превышает максимальную длину для строки OUT	
	Максимальная длина IN1, IN2 или OUT не соответствует назначенной области памяти	
	Максимальная длина строк IN1 или IN2 равна 255 или максимальная длина строки OUT равна 0 или 255 (тип данных String)	
	Максимальная длина строк IN1 или IN2 равна 65534 или максимальная длина строки OUT равна 0 или 65534 (тип данных WString)	

9.2.4.8 FIND (поиск символов в строке символов)

Таблица 9-67 Инструкция поиска части строки

LAD/FBD	SCL	Описание

	`out := FIND(`
	`  in1:=_string_in_,`
	`  in2:=_string_in);`
		Выводит позицию указанной в IN2 строки символов в символьной строке IN1. Поиск начинается слева. Позиция первого совпадения IN2 выводится в OUT. Если строка символов IN2 отсутствует в символьной строке IN1, то выводится ноль.

Таблица 9-68 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
IN1	IN	String, WString	Искать в этой символьной строке
IN2	IN	String, WString	Искать эту строку символов
OUT	OUT	Int	Позиция в символьной строке IN1, первый положительный результат поиска

Таблица 9-69 Состояние ENO

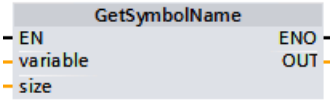
ENO	Условие	OUT
1	Ошибки отсутствуют	Действительная позиция символа
0	IN2 больше, чем IN1	Позиция символа устанавливается на 0
	Фактическая длина IN1 превышает максимальную длину IN1 или фактическая длина IN2 превышает максимальную длину IN2 (неправильная строка)	
	Максимальная длина IN1 или IN2 не соответствует назначенной области памяти	
	Максимальная длина IN1 или IN2 равна 255 (тип данных String) или 65535 (тип данных WString)	

9.2.5 Информация при выполнении

9.2.5.1 GetSymbolName (считывание имени переменной во входном параметре)

Таблица 9-70 Инструкция GetSymbolName

LAD/FBD	SCL	Описание

	`OUT := GetSymbolName(`
	`  variable:=_parameter_in_,`
	`  size:=_dint_in_);`
		Инструкция GetSymbolName выводит строку символов, соответствующую имени переменной из интерфейса блока.
		Инструкция может вызываться в программе многократно с различными переменными. Технологическое значение переменной не играет роли.
		Инструкция выводит считанное имя на параметре OUT.

Параметр

В приведенной ниже таблице показаны параметры инструкции GetSymbolName:

Параметр	Объявление	Тип данных	Область памяти	Описание
VARIABLE	Input	PARAMETER	Части параметра Input, Output, InOut	Переменная из локального интерфейса блока, имя которой должно быть выведено как строковое значение
SIZE	Input	DINT	I, Q, M, D, L	Ограничивает число символов на вывод в параметре OUT: • SIZE > 0: GetSymbolName выводит первые SIZE символы имени. • SIZE = 0: GetSymbolName выводит полное имя. • SIZE < 0: GetSymbolName выводит последние SIZE символы имени.
OUT	Return	WSTRING	I, Q, M, D, L	Вывод представленного входным параметром имени переменной (тега)

Входные параметры интерфейса блока указываются в параметре VARIABLE. Следует использовать только интерфейсный параметр для этого параметра, а не переменную (тег) PLC или переменную (тег) блока данных.

Для ограничения длины считанного имени переменной (тега) следует использовать параметр SIZE. Если инструкция обрезает имя, то это ясно видно по символу "..." (Unicode символ 16#2026) на конце имени. Следует помнить, что длина этого символа 1.

Дополнительную информацию о допустимых типах данных можно найти в разделе "Типы данных (Страница 123)".

Пример: Значение параметра SIZE

Следующий пример поясняет значение параметра SIZE. Следующее имя переменной (тега) считывается из интерфейса блока: "MyPLCTag" (двойные кавычки в начале и конце имени являются его частью).

SIZE	Вывод GetSymbolName	Значение
1	'...'	• Первый символ WSTRING:' • Признак того, что имя было обрезано: ... • Последний символ WSTRING:'
2	'"...'	• Первый символ WSTRING:' • Первый символ имени и признак того, что имя было обрезано:"... • Последний символ WSTRING:'
3	'"M...'	• Первый символ WSTRING:' • Первые два символа имени и признак того, что имя было обрезано:"... "M... • Последний символ WSTRING:'

SIZE	Вывод GetSymbolName	Значение
6	'"MyPL...'	• Первый символ WSTRING:' • Первые пять символов имени и признак того, что имя было обрезано: "MyPL... • Последний символ WSTRING:'
0	'"MyPLCTag"'	• Первый символ WSTRING:' • Все символы имени: "MyPLCTag" • Последний символ WSTRING:'

Пример: Чтение символьного имени

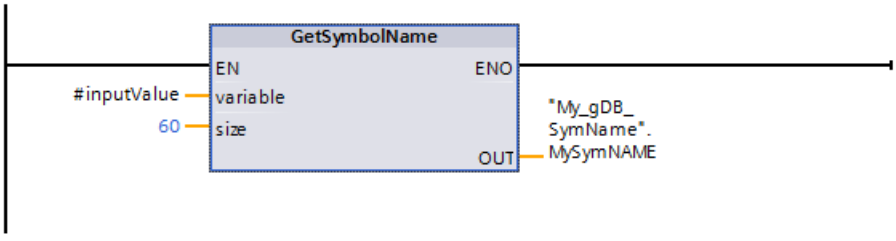
В следующем примере считывается имя переменной (тега), подсоединенной через входной параметр блока.

Создать две переменные (теги) в глобальном блоке данных для хранение данных.

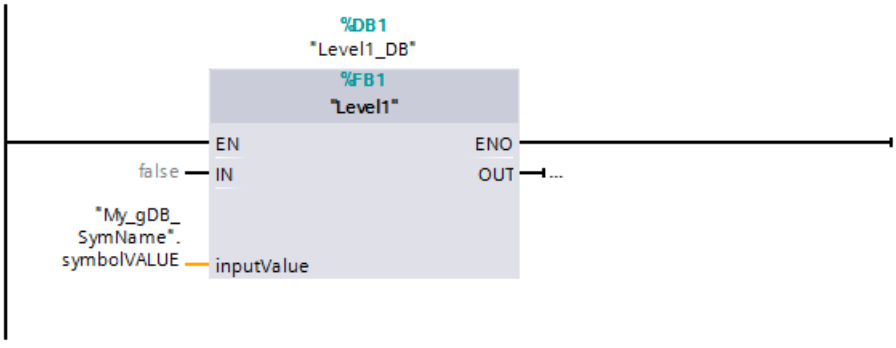
My_gDB_SymName

		Name	Data type	Start value
1		Static		
2		MySymNAME	WString	WSTRING#''
3		symbolVALUE	Byte	16#42

Создать входной параметр inputValue с типом данных BYTE в блоке Level1. Вызвать инструкцию GetSymbolName в блоке Level1. Подсоединить параметры инструкции следующим образом:



Подсоединить параметр inputValue блока Level1 следующим образом:



Инструкция GetSymbolName выполняется в блоке Level1. Входной параметр inputValue блока Level1 проверяется по входному параметру VARIABLE инструкции на предмет его соединения. При этом считывается переменная (тег) symbolVALUE и выводится на

выходном параметре OUT как строка символов ("MySymNAME"). Согласно значению входного параметра SIZE, длина символьной строки ограничивается 60 символами.

My_gDB_SymName

		Name	Data type	Start value	Monitor value
1		Static			
2		MySymNAME	WString	WSTRING#''	WSTRING#'"My_gDB_SymName".symbolVALUE'
3		symbolVALUE	Byte	16#42	16#42

9.2.5.2 GetSymbolPath (запрос составного глобального имени входного параметра)

Таблица 9-71 Инструкция GetSymbolPath

LAD/FBD	SCL	Описание

GetSymbolPath
EN — ENO
variable — OUT
size
	`OUT := GetSymbolPath(`
	`  variable:=_parameter_in_,`
	`  size:=_dint_in_);`
		Инструкция GetSymbolPath считывает составное глобальное имя входного параметра на локальном интерфейсе блока (FB или FC). Имя состоит из пути доступа и имени переменной (тега).
		Инструкция может вызываться в программе многократно с различными переменными. Технологическое значение переменной не играет роли.
		Инструкция выводит считанное имя на параметре OUT.

Параметр

В приведенной ниже таблице показаны параметры инструкции GetSymbolPath:

Параметр	Объявление	Тип данных	Область памяти	Описание
VARIABLE	Input	PARAMETER	Части параметра Input, Output, InOut	Выбор локального интерфейса, для которого необходимо прочитать глобальное имя источника входного параметра.
SIZE	Input	DINT	I, Q, M, D, L или постоянная	Ограничивает число символов на вывод в параметре OUT. • SIZE > 0: GetSymbolPath выводит первые SIZE символы имени. • SIZE = 0: GetSymbolPath выводит полное имя. • SIZE < 0: GetSymbolPath выводит последние SIZE символы имени.
OUT	Output	WSTRING	I, Q, M, D, L	Вывод имени переменной (тега) источника входного параметра.

Дополнительную информацию о допустимых типах данных можно найти в разделе "Типы данных (Страница 123)".

Использование

Обратить внимание на следующие советы по использованию инструкции GetSymbolPath:

- Указать интерфейс блока, через который считывается имя входной переменной (тега), в параметре VARIABLE инструкции:
  - Если переменная (тег) предоставляет входной параметр, то GetSymbolPath указывает имя DB, его структуры и имя переменной (тега).
  - Если переменная (тег) PLC предоставляет входной параметр, то GetSymbolPath выводит имя переменной (тега) PLC.
  - Если константа предоставляет входной параметр, то GetSymbolPath выводит значение константы.
- Для ограничения длины считанного имени переменной (тега) следует использовать параметр SIZE. Если имя было обрезано, то это ясно видно по символу "..." (Unicode символ 16#2026) на конце имени. Следует помнить, что длина этого символа 1.

Пример: Значение параметра SIZE

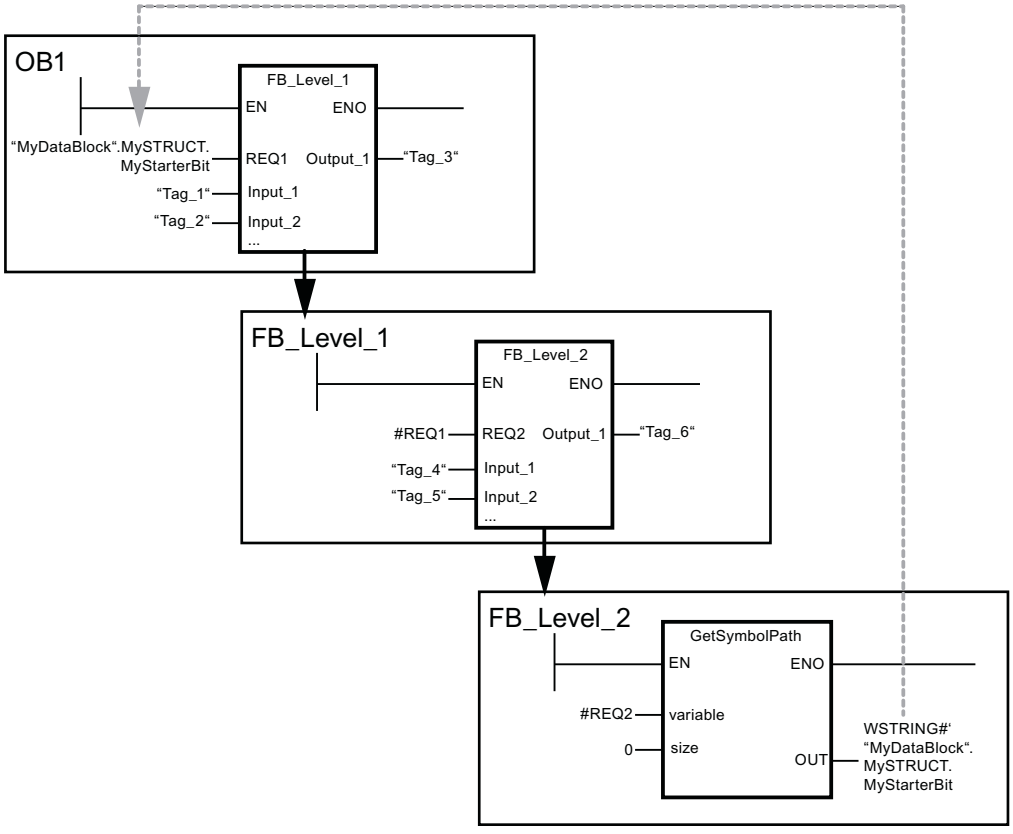
Следующий пример поясняет значение параметра SIZE. Инструкция GetSymbolPath считала следующее имя переменной (тега) из интерфейса блока: "MyPLCTag" (двойные кавычки в начале и конце имени являются его частью).

SIZE	Вывод GetSymbolPath	Значение
1	'...'	• Первый символ WSTRING:' • Признак того, что имя было обрезано: ... • Последний символ WSTRING:'
2	'"...'	• Первый символ WSTRING:' • Первый символ имени и признак того, что имя было обрезано:"... • Последний символ WSTRING:'
3	'"M...'	• Первый символ WSTRING:' • Первые два символа имени и признак того, что имя было обрезано:"... "M... • Последний символ WSTRING:'
6	'"MyPL...'	• Первый символ WSTRING:' • Первые пять символов имени и признак того, что имя было обрезано: "MyPL... • Последний символ WSTRING:'
0	'"MyPLCTag"'	• Первый символ WSTRING:' • Все символы имени: "MyPLCTag" • Последний символ WSTRING:'

Пример: Вызов GetSymbolPath по нескольким уровням вызовов блока

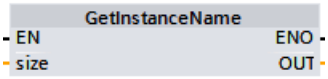
В следующем примере показано использование GetSymbolPath по нескольким уровням вызовов.

- Организационный блок OB1 вызывает блок FB_Level_1, который в свою очередь вызывает блок FB_Level_2.
- Блок FB_Level_2 выполняет GetSymbolPath, чтобы прочесть путь к параметру на интерфейсе REQ2.
- Так как интерфейс REQ1 выводит значение REQ2 инструкция определяет путь входного параметра REQ1.
- Переменная (тег) MyStarterBit - это входной параметр REQ1. Бит находится в структуре MySTRUCT в блоке данных MyDatablock.
  GetSymbolPath считывает эту информацию и выводит путь ("MyDataBlock".MySTRUCT.MyStarterBit) на параметр OUT.



9.2.5.3 GetInstanceName (считывание имени экземпляра блока)

Таблица 9-72 Инструкция GetInstanceName

LAD/FBD	SCL	Описание
	`OUT := GetInstanceName(` `  size:=_dint_in_);`	С помощью инструкции GetInstanceName можно считать имя блока данных экземпляра в функциональном блоке.

Параметр

В приведенной ниже таблице показаны параметры инструкции GetInstanceName:

Параметр	Объявление	Тип данных	Область памяти	Описание
SIZE	Input	DINT	I, Q, M, D, L или постоянная	Ограничивает число символов на вывод в параметре OUT. • SIZE > 0: GetInstanceName выводит первые SIZE символы имени. • SIZE = 0: GetInstanceName выводит полное имя. • SIZE < 0: GetInstanceName выводит последние SIZE символы имени.
OUT	Output	WSTRING	I, Q, M, D, L	Считанное имя блока данных экземпляра

Дополнительную информацию о допустимых типах данных можно найти в разделе "Типы данных (Страница 123)".

Пример: Значение параметра SIZE

Для ограничения длины считанного имени экземпляра следует использовать параметр SIZE. Если инструкция обрезала имя, то это ясно видно по символу "..." (Unicode символ 16#2026) на конце имени. Следует помнить, что длина этого символа 1.

Следующий пример поясняет значение параметра SIZE. Инструкция GetInstanceName считала следующее имя экземпляра из интерфейса блока: "Level1_DB" (двойные кавычки в начале и конце имени являются его частью).

SIZE	Вывод GetSymbolPath	Значение
1	'...'	• Первый символ WSTRING:' • Признак того, что имя было обрезано: ... • Последний символ WSTRING:'
2	'"...'	• Первый символ WSTRING:' • Первый символ имени и признак того, что имя было обрезано:"... • Последний символ WSTRING:'
3	'"L...'	• Первый символ WSTRING:' • Первые два символа имени и признак того, что имя было обрезано:"... "L... • Последний символ WSTRING:'

SIZE	Вывод GetSymbolPath	Значение
6	'"Leve...'	• Первый символ WSTRING:' • Первые пять символов имени и признак того, что имя было обрезано: "Leve... • Последний символ WSTRING:'
0	'"Level1_DB"'	• Первый символ WSTRING:' • Все символы имени: "Level1_DB" • Последний символ WSTRING:'

Инструкция GetInstanceName записывает имя блока данных экземпляра в параметр OUT. Инструкция обрезает имя, если имя блока данных экземпляра превышает максимальную длину WSTRING.

Пример: Чтение имени блока данных экземпляра

Следующий пример показывает, как можно прочесть имя блока данных экземпляра.

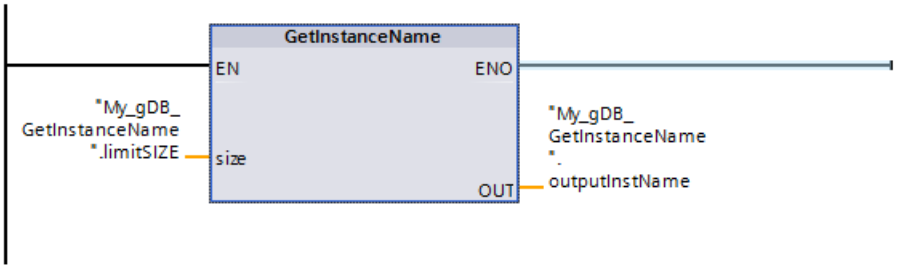
Создать две переменные (теги) в глобальном блоке данных для хранение данных.

Определить параметры инструкции следующим образом:

My_gDB_GetInstanceName

	Name	Datentyp	Startwert
1	Static		
2	limitSIZE	DInt	0
3	outputInsName	WString	WSTRING#''

Блок Level1_gin выполняет инструкцию GetInstanceName, при этом определяется соответствующий блок данных экземпляра блока Level1_gin и имя выводится на выходной параметр OUT как строка символов (outputInstName). Согласно значению 0 параметра SIZE (limitSIZE), длина символьной строки не ограничена.

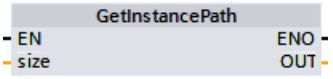


My_gDB_GetInstanceName

	Name	Data type	Start value	Monitor value
1	Static			
2	limitSIZE	DInt	0	0
3	outputInstName	WString	WSTRING#''	WSTRING#'"Level1_DB"'

9.2.5.4 GetSymbolPath (запрос составного глобального имени экземпляра блока)

Таблица 9-73 Инструкция GetInstancePath

LAD/FBD	SCL	Описание
	`OUT := GetInstancePath(` `  size:=_dint_in_);`	С помощью инструкции GetInstancePath выполняется чтение составного глобального имени экземпляра блока в функциональном блоке. Составленное глобальное имя экземпляра блока - это путь к полной иерархии вызовов, когда программа вызывает несколько экземпляров.

Параметр

В приведенной ниже таблице показаны параметры инструкции GetInstancePath:

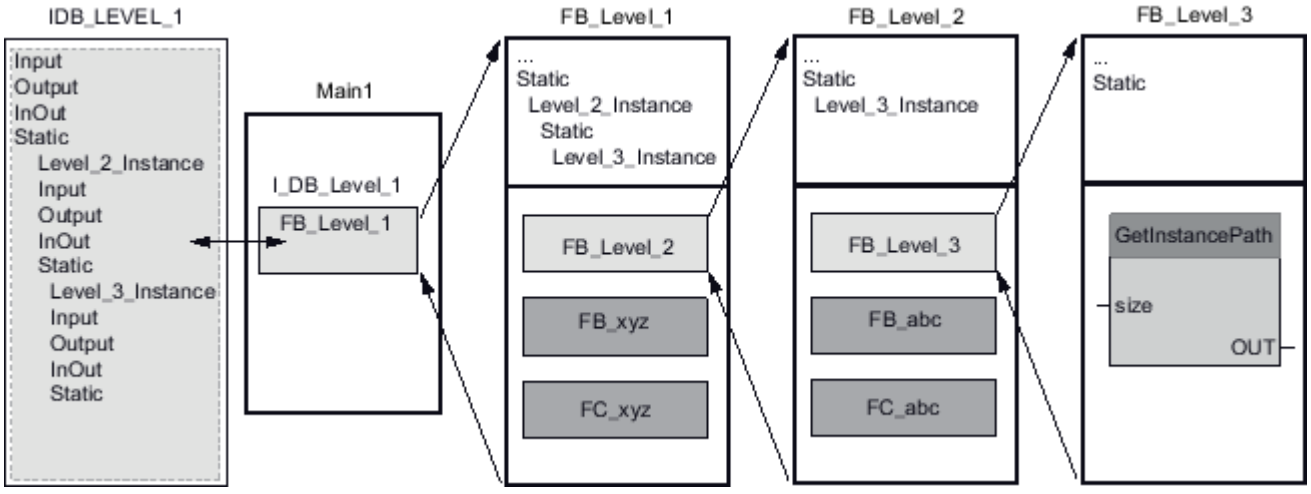
Параметр	Объявление	Тип данных	Область памяти	Описание
SIZE	Input	DINT	I, Q, M, D, L или постоянная	Ограничивает число символов на вывод в параметре OUT. • SIZE > 0: GetInstancePath выводит первые SIZE символы имени. • SIZE = 0: GeInstancePath выводит полное имя. • SIZE < 0: GetInstancePath выводит последние SIZE символы имени.
OUT	Output	WSTRING	I, Q, M, D, L	Считанное глобальное имя экземпляра блока Если глобальное имя экземпляра блока превышает максимальную длину WSTRING (254 символа), то GetInstancePath обрезает имя.

Дополнительную информацию о допустимых типах данных можно найти в разделе "Типы данных (Страница 123)".

Пример: Вызов GetInstancePath для запросы пути вызова FB с несколькими экземплярами

В следующем примере функциональный блок FB_Level_3 вызывает инструкцию GetInstancePath.

- Функциональный блок FB_Level_3 сохраняет свои данные в вызывающем функциональном блоке FB_Level_2.
- Функциональный блок FB_Level_2 в свою очередь сохраняет свои данные в вызывающем функциональном блоке FB_Level_1.
- Функциональный блок FB_Level_1 в свою очередь сохраняет свои данные в своем блоке данных экземпляра IDB_LEVEL_1. Благодаря использованию нескольких экземпляров, блок данных экземпляра FB_Level_1 содержит все данные трех функциональных блоков.



Инструкция GetInstancePath выводи в зависимости от значения параметра SIZE следующие значения для этого примера:

SIZE	Вывод GetInstancePath	Значение
1	'...'	• Первый символ WSTRING:' • Признак того, что имя было обрезано: ... • Последний символ WSTRING:'
2	'"...'	• Первый символ WSTRING:' • Первый символ имени и признак того, что имя было обрезано:"... • Последний символ WSTRING:'
3	'"I...'	• Первый символ WSTRING:' • Первые два символа имени и признак того, что имя было обрезано:"... "I... • Последний символ WSTRING:'
6	'"IDB_...'	• Первый символ WSTRING:' • Первые пять символов имени и признак того, что имя было обрезано: "IDB_... • Последний символ WSTRING:'
0	'"IDB_LEVEL_1".Level_2_ Instance.Level_3_Instance'	• Первый символ WSTRING:' • Все символы имени: "IDB_LEVEL_1".Level_2_Instance.Level_3_Instance • Последний символ WSTRING:'

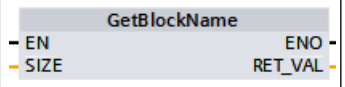
Примечание

Использование GetInstancePath в функциональных блоках с одним экземпляром

Если функциональный блок, в котором вызывается GetInstancePath, хранит данные в собственном блоке данных экземпляра, тоGetInstancePath выводит имя единственного экземпляра как глобальное имя. В этом случае результат на параметре OUT соответствует инструкции GetInstanceName (Страница 380).

9.2.5.5 GetBlockName (считывание имени блока)

Таблица 9-74 Инструкция GetBlockName

LAD/FBD	SCL	Описание
	`RET_VAL := GetBlockName( size:=_dint_in_);`	С помощью инструкции GetBlockName выполняется чтение имени блока, в котором вызывается инструкция.

Параметр

В приведенной ниже таблице показаны параметры инструкции GetBlockName:

Параметр	Объявление	Тип данных	Область памяти	Описание
SIZE	Input	UINT	I, Q, M, D, L или постоянная	Ограничивает число символов на вывод в параметре RET_VAL. • SIZE > 0: GetBlockName выводит первые SIZE символы имени. • SIZE = 0: GetBlockName выводит полное имя. • SIZE < 0: GetBlockName выводит последние SIZE символы имени.
RET_VAL	Output	WSTRING	I, Q, M, D, L	Считанное имя блока данных экземпляра

Дополнительную информацию о допустимых типах данных можно найти в разделе "Типы данных (Страница 123)".

Пример: Значение параметра SIZE

Для ограничения длины имени блока до определенного числа символов, следует указать макс. длину в параметре SIZE. Если инструкция GetBlockName обрезает имя, то это ясно видно по символу "..." (Unicode символ 16#2026) на конце имени. Следует помнить, что длина этого символа 1.

Следующий пример поясняет значение параметра SIZE. ИнструкцияGetBlockName считала следующее имя блока: Level1_gbn (двойные кавычки в начале и конце имени являются его частью).

SIZE	Вывод GetBlockName	Значение
1	'...'	• Первый символ WSTRING:' • Признак того, что имя было обрезано: ... • Последний символ WSTRING:'
2	'"...'	• Первый символ WSTRING:' • Первый символ имени и признак того, что имя было обрезано:"... • Последний символ WSTRING:'

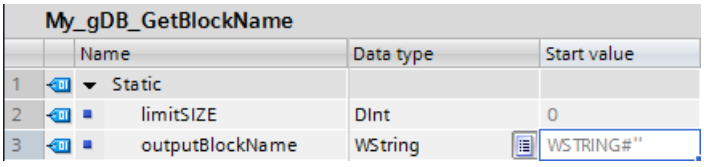
SIZE	Вывод GetBlockName	Значение
3	'"L...'	• Первый символ WSTRING:' • Первые два символа имени и признак того, что имя было обрезано:"... "L... • Последний символ WSTRING:'
6	'"Leve...'	• Первый символ WSTRING:' • Первые пять символов имени и признак того, что имя было обрезано: "Leve... • Последний символ WSTRING:'
0	'"Level1_gbn"'	• Первый символ WSTRING:' • Все символы имени: "Level1_gbn" • Последний символ WSTRING:'

Инструкция GetBlockName записывает имя блока в параметр RET_VAL. Если имя блока превышает максимальную длину WSTRING, то инструкция обрезает имя.

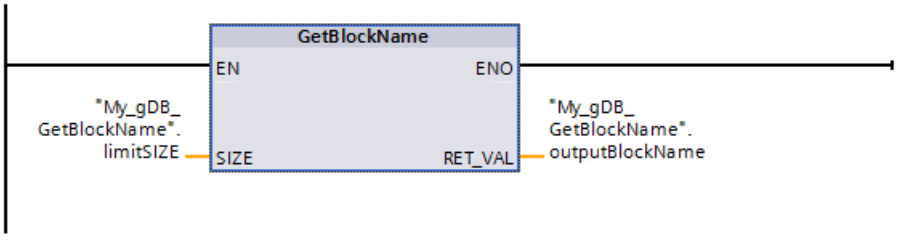
Пример: Чтение имени блока

В следующем примере показано, как считывается имя блока.

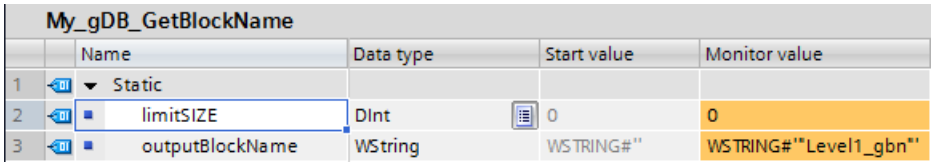
1. Создать две переменные (теги) в глобальном блоке данных для хранение данных.



2. Определить параметры инструкции следующим образом:



Блок Level1_gbn выполняет инструкцию GetBlockName. GetBlockName считывает имя блока Level1_gbn и имя выводится на выходной параметр RET_VAL как строка символов (outputBlockName). Так как параметр SIZE равен 0 (limitSIZE), длина символьной строки не ограничена.



9.3 Распределенный ввод/вывод (PROFINET, PROFIBUS или AS-i)

9.3.1 Инструкции для распределенного ввода/вывода

Следующие инструкции для распределенной периферии могут использоваться вместе с PROFINET, PROFIBUS или AS-i:

- Инструкция RDREC (Страница 387): Считывает набор данных с номером INDEX из модуля или устройства.
- Инструкция WRREC (Страница 387): Передает набор данных с номером INDEX в указанный ID модуль или устройство.
- Инструкция GETIO (Страница 390): Последовательно считывает все входы стандартного ведомого устройства DP / устройства ввода-вывода PROFINET.
- Инструкция SETIO (Страница 391): Последовательно передает данные из исходного диапазона, определенного параметром OUTPUTS, на адресованное стандартное ведомое устройство DP / устройство ввода-вывода PROFINET.
- Инструкция GETIO_PART (Страница 392): Последовательно считывает соответствующую часть входов модуля ввода-вывода.
- Инструкция SETIO_PART (Страница 394): Последовательно записывает данные из исходной области, указанной параметром OUTPUTS, на выходы модуля ввода-вывода.
- Инструкция RALRM (Страница 395): Позволяет получить прерывание со всей сопутствующей информацией от модуля или устройства и передать эту информацию в выходные параметры.
- Инструкция DPRD_DAT (Страница 409): Позволяет с помощью инструкции DPRD_DAT читать согласованные области данных размером больше 64 байт из модуля или устройства.
- Инструкция DPWR_DAT (Страница 409): Позволяет с помощью инструкции DPWR_DAT записывать согласованные области данных размером больше 64 байт в модуль или устройство.

Инструкция D_ACT_DP (Страница 399) позволяет целенаправленно отключать и включать сконфигурированные устройства PROFINET IO. Также можно определить, активировано или деактивировано каждое назначенное устройство PROFINET IO в настоящий момент.

Примечание

Примечание: Инструкция D_ACT_DP может использоваться только для устройств PROFINET IO. Нельзя использовать инструкцию с ведомыми устройствами PROFIBUS DP.

С помощью инструкции DPNRM_DG (Страница 417) могут быть считаны текущие диагностические данные ведомого устройства DP в формате, специфицированном в EN 50 170 Volume 2, PROFIBUS.

Примечание

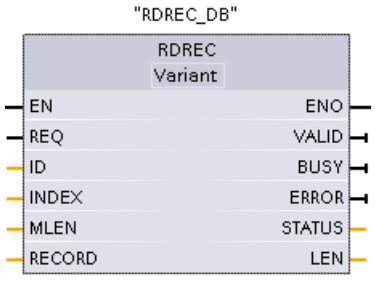
Инструкция DPNRM_DG может использоваться только для PROFIBUS .

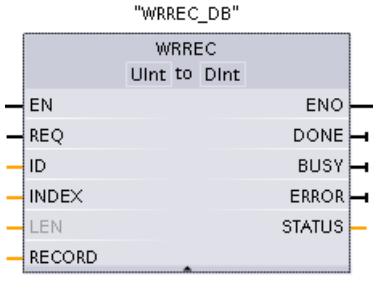
9.3.2 Инструкции RDREC и WRREC (чтение/запись набора данных)

Инструкции RDREC (чтение набора данных) и WRREC (запись набора данных) могут использоваться для PROFINET, PROFIBUS и AS-i.

Таблица 9-75 Инструкции RDREC и WRREC

LAD/FBD	SCL	Описание

	`"RDREC_DB"(`
	`req:=_bool_in_,`
	`ID:=_word_in_,`
	`index:=_dint_in_,`
	`mlen:=_uint_in_,`
	`valid=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_dword_out_,`
	`len=>_uint_out_,`
	
	`record:=_variant_inout_);`
		С помощью инструкции RDREC выполняется считывание набора данных с номером INDEX из компонента, указываемого через ID, например, из центральной монтажной стойки или распределенного компонента (PROFIBUS DP или PROFINET IO). Указать максимальное количество считываемых байтов в MLEN. Выбранная длина целевой области RECORD как минимум должна совпадать с длиной указанных MLEN байтов.

	`"WRREC_DB"(`
	`req:=_bool_in_,`
	`ID:=_word_in_,`
	`index:=_dint_in_,`
	`len:=_uint_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_dword_out_,`
	
	`record:=_variant_inout_);`
		С помощью инструкции WRREC выполняется передача набора данных (RECORD) с номером INDEX в компонент DP-Slave или устройства PROFINET IO, указываемых через ID, например, модуль в центральной монтажной стойке или распределенный компонент (PROFIBUS DP или PROFINET IO).
		Указать длину передаваемого набора данных в байтах. Поэтому выбранная длина исходной области RECORD как минимум должна совпадать с длиной указанных LEN байтов.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примерах SCL "RDREC_DB" и "WRREC_DB" - это имена DB экземпляра.

Таблица 9-76 Типы данных для параметров RDREC и WRREC

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	REQ = 1: Передача набора данных
ID	IN	HW_IO (Word)	Логический адрес ведомого устройства DP/компонента PROFINET IO (модуль или субмодуль): • Для модуля вывода необходимо установить бит 15 (например, для адреса 5: ID:=DW#16#8005). • Для комбинированного модуля должен быть указан меньший из двух адресов. **Примечание**: В версии 3.0 идентификатор устройства ID может быть определен двумя способами по выбору: • Путем выбора следующих опций на вкладке просмотра сетевых соединений: – Устройство (серое поле) – "Свойства" устройства – "Аппаратный идентификатор" **Примечание**: Аппаратный идентификатор отображается не для всех устройств. • Путем выбора следующих опций в дереве проекта: – Переменные (теги) PLC – Стандартная таблица переменных (тегов) – Вкладка "Константы" системы Отображаются аппаратные идентификаторы всех сконфигурированных устройств. **Примечание**: В версии 4.0 идентификатор устройства (аппаратный идентификатор) для интерфейсного модуля определяется в таблице переменных (тегов) через параметр "Имя устройства [HEAD]" в системных константах.
INDEX	IN	Byte, Word, USInt, UInt, SInt, Int, DInt	Номер набора данных
MLEN	IN	Byte, USInt, UInt	Максимальная длина в байтах запрашиваемой информации из набора данных (RDREC)
VALID	OUT	Bool	Новый набор данных принят и является действительным (RDREC). Бит VALID = TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок.
DONE	OUT	Bool	Набор данных был передан (WRREC). Бит DONE = TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок.
BUSY	OUT	Bool	• BUSY = 1: Процесс чтения (RDREC) или записи (WRREC) еще не завершен. • BUSY = 0: Передача набора данных завершена.
ERROR	OUT	Bool	ERROR = 1: Возникла ошибка чтения (RDREC) или ошибка записи (WRREC). Бит ERROR в течение одного цикла имеет значение TRUE, после того, как последний запрос был завершен с ошибкой. Код ошибки в параметре STATUS действителен только в цикле, где ERROR = TRUE.
STATUS	OUT	DWord	Состояние (Страница 404) блока или информация об ошибках (Страница 554)

Параметр и тип данных		Тип данных	Описание
LEN	OUT (RDREC) IN (WRREC)	UInt	• Длина запрошенной информации из набора данных (RDREC) • Максимальная длин передаваемого набора данных (WRREC) в байтах.
RECORD	IN_OUT	Variant	• Целевая область для запрошенного набора данных (RDREC) • Набор данных (WRREC)

Инструкции RDREC иWRREC работают асинхронно, т.е. обработка включает в себя несколько вызовов инструкций. Для запуска задания используется вызов RDREC или WRREC через REQ = 1.

Состояние задания отображается через выходной параметр BUSY и оба центральных байта выходного параметра STATUS. Передача набора данных завершена, если выходной параметр BUSY был установлен на значение FALSE.

Если выходной параметр VALID (RDREC) или DONE (WRREC) имеет значение TRUE (только в течение одного цикла), то это указывает на то, что набор данных был успешно передан в целевую область RECORD (RDREC) или на целевое устройство (WRREC). Для RDREC выходной параметр LEN содержит длину запрошенный данных в байтах.

Выходной параметр ERROR (только в течение одного цикла, если ERROR = TRUE) показывает, что при передаче набора данных возникла ошибка. В этом случае выходной параметр STATUS (только в течение одного цикла, в котором ERROR = TRUE) содержит информацию об ошибке.

Наборы данных определяются изготовителем оборудования. Для получения подробной информации о наборе данных следует обратиться к документации на конкретное устройство, предоставляемой его изготовителем.

Одновременно можно использовать до четырех инструкций RDREC и четырех инструкций WRREC.

Примечание

Если конфигурирование ведомого устройства DPV1 выполняется с помощью GSD-файла (GSD от версии 3), а DP-интерфейс ведущего устройства DP установлен на "совместимость с S7", то с помощью инструкции "RDREC" в программе пользователя считывание наборов данных из модулей ввода/вывода или запись наборов данных в модули ввода/вывода с помощью инструкции "WRREC" будет невозможной. В этом случае ведущее устройство DP адресует неправильный слот (сконфигурированный слот + 3).

Способ устранения: Установить "DPV1" в качестве интерфейса ведущего устройства DP.

Примечание

Интерфейсы инструкций "RDREC" и "WRREC" идентичны FB "RDREC" и "WRREC", определенным в "PROFIBUS Guideline PROFIBUS Communication and Proxy Function Blocks according to IEC 61131-3".

Примечание

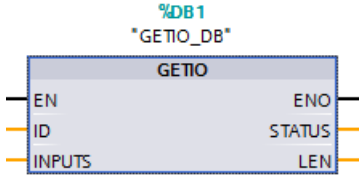
При использовании "RDREC" или "WRREC" для чтения или записи набора данных для PROFINET IO, модуль CPU интерпретирует отрицательные значения в параметрах INDEX, MLEN и LEN как 16-битные целые числа без знака.

9.3.3 GETIO (считывание образа процесса)

С помощью инструкции "GETIO" выполняется последовательное считывание входов модулей или субмодулей ведомых устройств DP и устройств PROFINET IO. Инструкция "GETIO" вызывает инструкцию "DPRD_DAT (Страница 409)". Если при передаче данных не возникли ошибки, то считанные данные записываются в целевую область, указанную в INPUTS.

Таблица 9-77 Инструкция GETIO (чтение образа процесса).

LAD/FBD	SCL	Описание

	`"GETIO_DB"(`
	`id:=_uint_in_,`
	`status=>_dword_out_,`
	`len=>_int_out_,`
	`inputs:=_variant_inout_);`
		С помощью инструкции "GETIO" последовательно считываются все входы стандартного ведомого устройства DP/устройства PROFINET IO.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "GETIO_DB" - это имя DB экземпляра.

Длина целевой области должна быть больше или равна длине выбранного компонента.

При чтении из стандартного ведомого устройства DP с модульной структурой или с несколькими идентификаторами DP, при каждом вызове инструкции "GETIO" происходит обращение только к данным одного компонента/идентификатора DP по сконфигурированному начальному адресу.

Параметр

В приведенной ниже таблице показаны параметры инструкции "GETIO":

Параметр	Объявление	Тип данных	Описание
ID	IN	HW_SUBMODULE	Аппаратный идентификатор стандартного ведомого устройства DP/устройства PROFINET IO
STATUS[1]	OUT	DWord	Содержит информацию об ошибке "DPRD_DAT (Страница 409)" в формате DW#16#40xxxx00.

Параметр	Объявление	Тип данных	Описание
LEN	OUT	Int	Объем считанных данных в байтах
INPUTS	IN_OUT	Variant	Целевая область для считанных данных: Длина целевой области должна быть больше или равна длине выбранного стандартного ведомого устройства DP/устройства PROFINET IO. Могут использоваться следующие типы данных: • Типы системных данных и массив типов системных данных: BYTE, CHAR, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL, LREAL, LWORD, LINT • Типы данных, определенные пользователем (UDT) • Структуры (STRUCT), но только не в оптимизированном блоке данных (DB)

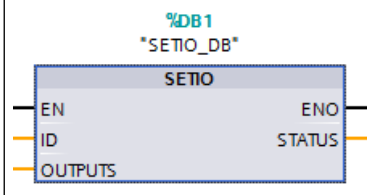
[1] Для отображения кодов ошибок GETIO следует использовать тип данных DWord.

9.3.4 SETIO (передача образа процесса)

С помощью инструкции "SETIO" выполняется последовательная передача данных из определенной параметром OUTPUTS исходной области в адресованные модули или субмодули ведомых устройств DP и устройств PROFINET IO. Если соответствующий диапазон адресов стандартного ведомого устройства DP/устройства PROFINET IO был сконфигурирован как цельная область в образе процесса, то данные передаются в образ процесса. Инструкция "SETIO" при этой передаче вызывает инструкцию "DPWR_DAT (Страница 409)".

Таблица 9-78 Инструкция SETIO (передача образа процесса)

LAD/FBD	SCL	Описание

	`"SETIO_DB"(`
	`  id:=_uint_in_,`
	`  status=>_dword_out_,`
	`  outputs:=_variant_inout_);`
		С помощью инструкции "SETIO" выполняется последовательная передача данных из исходного диапазона, определенного параметром OUTPUTS, на адресованное стандартное ведомое устройство DP / устройство PROFINET IO.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "SETIO_DB" - это имя DB экземпляра.

Длина исходной области должна быть больше или равна длине выбранного компонента.

В случае стандартного ведомого устройства DP/устройства PROFINET IO с модульной структурой или с несколькими идентификаторами DP, при каждом вызове SETIO можно обращаться только к одному идентификатору DP/компоненту.

Параметр

В приведенной ниже таблице показаны параметры инструкции "SETIO":

Параметр	Объявление	Тип данных	Описание
ID	IN	HW_SUBMODULE	Аппаратный идентификатор стандартного ведомого устройства DP/устройства PROFINET IO
STATUS[1]	OUT	DWord	Содержит информацию об ошибке "DPWR_DAT (Страница 409)" в формате DW#16#40xxxx00.
OUTPUTS	IN_OUT	Variant	Исходная область для записываемых данных: Длина исходной области должна быть больше или равна длине выбранного стандартного ведомого устройства DP/устройства PROFINET IO. Могут использоваться следующие типы данных: • Типы системных данных и массив типов системных данных: BYTE, CHAR, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL, LREAL, LWORD, LINT • Типы данных, определенные пользователем (UDT) • Структуры (STRUCT), но только не в оптимизированном блоке данных (DB)

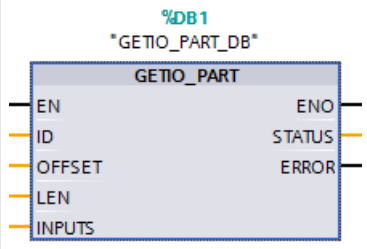
[1] Для отображения кодов ошибок SETIO следует использовать тип данных DWord.

9.3.5 GETIO_PART (считывание области образа процесса)

С помощью инструкции "GETIO_PART" выполняется последовательное считывание соответствующей части входов модулей или субмодулей ведомых устройств DP и устройств PROFINET IO. Инструкция GETIO_PART вызывает инструкцию "DPRD_DAT (Страница 409)".

Таблица 9-79 Инструкция GETIO_PART (чтение области образа процесса)

LAD/FBD	SCL	Описание

	`"GETIO_PART_DB"(`
	`id:=_uint_in_,`
	`offset:=_int_in_,`
	`len:=_int_in_,`
	`status=>_dword_out_,`
	`error=>_bool_out_,`
	`inputs:=_variant_inout_);`
		Инструкция GETIO_PART обеспечивает последовательное считывание соответствующей части входов модуля ввода-вывода.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "GETIO_PART_DB" - это имя DB экземпляра.

С входным параметром ID для выбора IO-модуля используется аппаратный ID.

С помощью параметров OFFSET и LEN указывается часть образа процесса для считывания. Если указанная OFFSET und LEN входная область не полностью охватывается модулем, то блок возвращает код ошибки DW#16#40**80B7**00.

Длина целевой области должна быть больше или равна числу считываемых байтов:

- Если при передаче данных не возникли ошибки, то ERROR принимает значение FALSE. Считанные данные записываются в определенную параметром INPUTS целевую область.
- Если при передаче данных возникает ошибка, то ERROR принимает значение TRUE. Параметр STATUS получает информацию об ошибках из DPRD_DAT.
- Если целевая область больше LEN, то инструкция выполняет запись в первые байты LEN целевой области. ERROR принимает значение FALSE.

Параметр

В приведенной ниже таблице показаны параметры инструкции GETIO_PART:

Параметр	Объявление	Тип данных	Описание
ID	IN	HW_SUBMODULE	Аппаратный идентификатор модуля
OFFSET	IN	Int	Номер первого считываемого в образе процесса байта для компонента (наименьшее возможное значение: 0)
LEN	IN	Int	Количество считываемых байт
STATUS[1]	OUT	DWord	Содержит информацию об ошибке "DPRD_DAT (Страница 409)" в формате DW#16#40xxxx00, если ERROR = TRUE.
ERROR	OUT	Bool	Индикация ошибок: ERROR = TRUE, если при вызове DPRD_DAT (Страница 409) возникает ошибка
INPUTS	IN_OUT	Variant	Целевая область для считанных данных: Если целевая область больше LEN, то инструкция выполняет запись в первые байты LEN целевой области. Могут использоваться следующие типы данных: • Типы системных данных и массив типов системных данных: BYTE, CHAR, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL, LREAL, LWORD, LINT • Типы данных, определенные пользователем (UDT) • Структуры (STRUCT), но только не в оптимизированных блоках данных (DB)

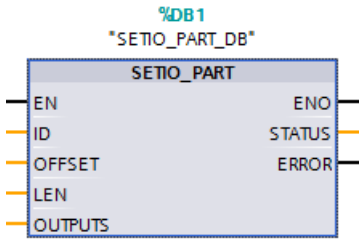
[1] Для отображения кодов ошибок GETIO_PART следует использовать тип данных DWord.

9.3.6 SETIO_PART (передача области образа процесса)

С помощью инструкции "SETIO_PART" выполняется последовательная запись данных из определенной параметром OUTPUTS исходной области на выходы модулей и субмодулей ведомых устройств DP и устройств PROFINET IO. SETIO_PART вызывает инструкцию "DPWR_DAT (Страница 409)".

Таблица 9-80 Инструкция SETIO_PART (передача области образа процесса)

LAD/FBD	SCL	Описание

	`"SETIO_PART_DB"(`
	`id:=_uint_in_,`
	`offset:=_int_in_,`
	`len:=_int_in_,`
	`status=>_dword_out_,`
	`error=>_bool_out_,`
	`outputs:=_variant_inout_);`
		С помощью инструкции SETIO_PART выполняется последовательная запись данных из исходной области, указанной параметром OUTPUTS, на выходы модуля ввода-вывода.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "SETIO_PART_DB" - это имя DB экземпляра.

С входным параметром ID, для выбора IO-модуля используется указанное оборудование.

С помощью параметров OFFSET и LEN, назначается часть образа процесса, которая должна быть записана для адресованного ID компонента. Если указанная OFFSET und LEN выходная область не полностью охватывается модулем, то блок возвращает код ошибки DW#16#40**80B7**00.

Длина целевой области должна быть больше или равна числу считываемых байтов:

- Если при передаче данных не возникли ошибки, то ERROR принимает значение FALSE.
- Если при передаче данных возникает ошибка, то ERROR принимает значение TRUE и STATUS получает информацию об ошибке из DPWR_DAT.
- Если исходная область больше LEN, то инструкция передает первые байты LEN из OUTPUTS. ERROR принимает значение FALSE.

Параметр

В приведенной ниже таблице показаны параметры инструкции SETIO_PART:

Параметр	Объявление	Тип данных	Описание
ID	IN	HW_SUBMODULE	Аппаратный идентификатор модуля ввода-вывода
OFFSET	IN	Int	Номер первого записываемого в образе процесса байта для компонента (наименьшее возможное значение: 0)
LEN	IN	Int	Количество записываемых байт
STATUS[1]	OUT	DWord	Содержит информацию об ошибке из "DPWR_DAT (Страница 409)" в формате DW#16#40xxxx00, если ERROR = TRUE

Параметр	Объявление	Тип данных	Описание
ERROR	OUT	Bool	Индикация ошибок: ERROR = TRUE, если при вызове DPWR_DAT (Страница 409) возникает ошибка
OUTPUTS	IN_OUT	Variant	Исходная область для записываемых данных: Если исходная область больше LEN, то передаются первые байты LEN из OUTPUTS. Могут использоваться следующие типы данных: • Типы системных данных и массив типов системных данных: BYTE, CHAR, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL, LREAL, LWORD, LINT • Типы данных, определенные пользователем (UDT) • Структуры (STRUCT), но только не в оптимизированных блоках данных (DB)

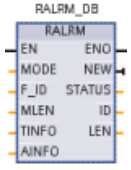
[1] Для отображения кодов ошибок SETIO_PART следует использовать тип данных DWord.

9.3.7 Инструкция RALRM (получение прерывания)

Инструкция RALRM (чтение прерывания) может использоваться для PROFINET и PROFIBUS .

Таблица 9-81 Инструкция RALRM

LAD/FBD	SCL	Описание

	`"RALRM_DB"(`
	`mode:=_int_in_,`
	`f_ID:=_word_in_,`
	`mlen:=_uint_in_,`
	`new=>_bool_out_,`
	`status=>_dword_out_,`
	`ID=>_word_out_,`
	`len=>_uint_out_,`
	`tinfo:=_variant_inout_,`
	`ainfo:=_variant_inout_);`
		С помощью инструкции RALRM (чтение прерывания) выполняется считывание диагностической информации из PROFIBUS- или PROFINET I/O-модуля/устройства.
		В информации в выходных параметрах содержаться сведения как о запуске вызванного блока OB, так и сведения из источника прерывания.
		Вызвать RALRM в OB прерываний, чтобы вывести информацию о событиях, вызвавших прерывание. S7-1200 поддерживает следующие OB диагностических прерываний: состояние, обновление, профиль, диагностическое прерывание, вставка или удаление модулей, отказ стойки или станции

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "RALRM_DB" - это имя DB экземпляра.

Таблица 9-82 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
MODE	IN	Byte, USInt, SInt, Int	Рабочее состояние
F_ID	IN	HW_IO (Word)	Логический начальный адрес компонента (модуля), с которого должны приниматься прерывания. **Примечание**: Идентификатор устройства ID может быть определен двумя способами по выбору: • Путем выбора следующих опций на вкладке просмотра сетевых соединений: – Устройство (серое поле) – "Свойства" устройства – "Аппаратный идентификатор" **Примечание**: Аппаратный идентификатор отображается не для всех устройств. • Путем выбора следующих опций в дереве проекта: – Переменные (теги) PLC – Стандартная таблица переменных (тегов) – Вкладка "Константы" системы – Отображаются аппаратные идентификаторы всех сконфигурированных устройств.
MLEN	IN	Byte, USInt, UInt	Максимальная длина в байтах принимаемой информации из набора данных. Если MLEN равен 0, то может быть получено столько информации о прерывании, сколько вмещает целевая область AINFO.
NEW	OUT	Bool	Принято новое прерывание.
STATUS	OUT	DWord	Состояние инструкции RALRM. Дополнительную информацию можно найти здесь: "Параметр STATUS для RDREC, WRREC и RALRM" (Страница 404).
ID	OUT	HW_IO (Word)	Аппаратный идентификатор модуля ввода/вывода, вызвавшего диагностическое прерывание **Примечание**: В описании параметра F_ID можно найти объяснение того, как определяется ID устройства.
LEN	OUT	DWord, UInt, UDInt, DInt, Real, LReal	Длина полученной информации о прерывании AINFO
TINFO	IN_OUT	Variant	Информация о задании: Целевая область для пусковой и административной информации OB. Длина TINFO всегда 32 байта.
AINFO	IN_OUT	Variant	Информация о прерывании: Целевая область для содержимого заголовка и дополнительной информации о прерывании. Для AINFOдолжна быть указана длина, как минимум соответствующая указанному в MLENчислу байт, если MLEN больше 0. Длина AINFO является переменной.

Примечание

Если инструкция "RALRM" вызывается в OB, пусковое событие которого не является прерыванием I/O, инструкция предоставит на своих выходах соответственно меньший объем информации.

Следует убедиться, что используются разные DB экземпляра при вызове "RALRM" в разных OB. При обработке данных из вызова "RALRM" вне соответствующего OB прерывания, для каждого пускового события OB следует использовать свой DB экземпляра.

Примечание

Интерфейс инструкций "RALRM" идентичен FB "RALRM", определенному в "PROFIBUS Guideline PROFIBUS Communication and Proxy Function Blocks according to IEC 61131-3".

Вызов RALRM

Инструкцию RALRM можно вызывать в трех различных режимах работы (MODE).

Таблица 9-83 Режим работы инструкции RALRM

РЕЖИМ	Описание
0	• ID содержит аппаратный идентификатор модуля ввода/вывода, вызвавшего прерывание. • Выходной параметр NEW устанавливается на TRUE. • LEN создает выход 0. • AINFO и TINFO не получают новой информации.
1	• ID содержит аппаратный идентификатор модуля ввода/вывода, вызвавшего прерывание. • Выходной параметр NEW устанавливается на TRUE. • LEN создает выход с размером по числу байт выведенных данных AINFO. • AINFO и TINFO обновляются с информацией о прерывании.
2	Если аппаратный идентификатор входного параметра F_ID вызвал прерывание, то действует следующее: • ID содержит аппаратный идентификатор модуля ввода/вывода, вызвавшего прерывание. Это значение должно совпадать со значением F_ID. • Выходной параметр NEW устанавливается на TRUE. • LEN создает выход с размером по числу байт выведенных данных AINFO. • AINFO и TINFO обновляются с информацией о прерывании.

Примечание

При назначении слишком короткой целевой области для TINFO илиAINFO, RALRM не может вернуть всю информацию.

MLEN может ограничить выводимый объем данных AINFO.

Информацию об интерпретации данных TINFO и AINFO можно найти в параметрах AINFO и TINFO в интерактивной информационной системе STEP 7.

Данные организационного блока TInfo

В нижеследующей таблице показано распределение данных TInfo для инструкции RALRM:

Также для OB: состояние, обновление, профиль, диагностическое прерывание, вставка или удаление модулей, отказ стойки или станции	0	Формат SI	Класс OB	№ OB	
	4	LADDR			
TI_Submodule - OB: Состояние, обновление, профиль	4			Слот	
	8	Спецификатор		0	
TI_DiagnosticInterrupt - OB: Диагностическое прерывание	4			Состояние I/O	
	8	Канал		MultiError	0
TI_PlugPullModule - OB: Удаление или вставка модулей	4			Класс события	ID ошибки
	8	0		0	
TI_StationFailure - OB: Отказ стойки или станции	4			Класс события	ID ошибки
	8	0		0	
Также для OB: состояние, обновление, профиль, диагностическое прерывание, вставка или удаление модулей, отказ стойки или станции	12	0			
	16				
	20	Адрес		slv_prfl	intr_type
	24	Флаги1	Флаги2	ID	
	28[1]	Изготовитель		Экземпляр	

[1] Байты 28 - 31 (изготовитель и экземпляр) не используется для PROFIBUS.

Примечание

Подробную информацию по данным TINFO можно найти в интерактивной информационной системе STEP 7.

9.3.8 D_ACT_DP (деактивация/активация ведомых устройств DP)

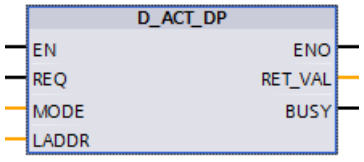
С помощью инструкции "D_ACT_DP" возможна целевая активация и деактивация сконфигурированных устройств PROFINET IO. Также можно определить, активировано или деактивировано каждое назначенное устройство PROFINET IO в настоящий момент.

Примечание

Инструкция D_ACT_DP может использоваться только для устройств PROFINET IO. Нельзя использовать инструкцию с ведомыми устройствами PROFIBUS DP.

Таблица 9-84 Инструкция D_ACT_DP

LAD/FBD	SCL	Описание

	`"D_ACT_DP_DB"(`
	`req:=_bool_in_,`
	`mode:=_usint_in_,`
	`laddr:=_uint_in_,`
	`ret_val=>_int_out_,`
	`busy=>_bool_out_);`
		С помощью инструкции D_ACT_DP можно включить или выключить сконфигурированное устройство PROFINET IO и определить, активировано или деактивировано назначенное устройство PROFINET IO в настоящий момент.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "D_ACT_DP_SFB_DB" - это имя DB экземпляра.

Шлюз типа IE/PB Link PN IO нельзя активировать/деактивировать с помощью инструкции D_ACT_DP. При использовании D_ACT_DP для названного шлюза, CPU возвращает значение W#16#8093 (нет аппаратного объекта, который может быть активирован или деактивирован по указанному в LADDR).

Примечание

Задание деактивации или активации должно несколько раз пройти через контрольную точку цикла. Поэтому нельзя ожидать завершения такого задания в запрограммированном цикле.

Описание функции

D_ACT_DP - это асинхронная инструкция, т.е. обработка задания растягивается на несколько прогонов инструкции D_ACT_DP. Для запуска задания используется вызов D_ACT_DP с REQ = 1.

Выходные параметры RET_VAL и BUSY сообщают о состоянии задания.

Применение

Если в модуле CPU конфигурируется устройство PROFINET IO, отсутствующее в действительности или ненужное в настоящий момент, то CPU все же продолжает регулярно обращаться к этому устройству PROFINET IO. После деактивации этих устройств обращения CPU прекращаются. Соответствующие события ошибок больше появляться не будут.

Примеры

С точки зрения машиностроителя, при серийном производстве оборудования существует множество возможных опций устройств. Однако каждая поставляемая машина имеет только индивидуальную комбинацию выбранных опций.

Изготовитель конфигурирует каждую из таких возможных опций машины как устройство PROFINET IO. Он это делает для того, чтобы разработать и поддерживать общую программу пользователя со всеми возможными опциями. С помощью D_ACT_DP отключаются все устройства PROFINET IO, отсутствующие при запуске машины.

Аналогичная ситуация характерна и для станков с различными опциями для инструмента, из которых постоянно используется только меньшая часть. Такие инструменты реализованы как устройства PROFINET IO. С помощью D_ACT_DP программа пользователя включает инструменты, необходимые в настоящий момент, и отключает те их них, которые потребуются в будущем.

Идентификация задания

Если задание по деактивации или активации было запущено и снова вызывается инструкция D_ACT_DP, до полного завершения задания, то поведение инструкции зависит от того, относится ли новый вызов к тому же заданию или нет. Если входной параметр LADDR тот же, то вызов обрабатывается как повторный вызов (follow-on).

Деактивация устройства PROFINET IO

При деактивации устройства PROFINET IO черезD_ACT_DP, технологические выходы устройства устанавливаются на сконфигурированные замещающие значения или на 0 (безопасное состояние). Назначенный контроллер PROFINET IO перестает обращаться к этим компонентам. Светодиоды ошибки на контроллере PROFINET IO или на CPU не идентифицируют деактивированные устройства PROFINET IO как находящиеся в состоянии ошибки или отсутствующие.

CPU обновляет образ процесса для входов деактивированных устройств PROFINET IO с 0. Поэтому CPU обрабатывает деактивированные устройства PROFINET IO точно так же, как неисправные устройства PROFINET IO.

При прямом обращении к данным пользователя прежде деактивированного устройства PROFINET IO через программу, поведение системы зависит от выбранного для блока способа обработки ошибок:

- Если активирована глобальная обработка ошибок, то система вносит стартовое событие ошибки доступа в буфер диагностики и остается в рабочем состоянии RUN.
- Если активирована локальная поблочная обработка ошибок, то система вносит причину ошибки в структуру ошибок. Причину ошибки можно запросить с помощью инструкции GET_ERROR_ID (Страница 320).
  Ошибка при доступе по чтению возвращает значение 0. Дополнительная информация по обработке ошибок см. "Приоритеты и лист ожидания для обработки событий" (Страница 103).

При попытке обращения к деактивированному устройству PROFINET IO через инструкцию (напр., RD_REC (Страница 387)), в RET_VAL появляется информация об ошибке, аналогичная информации, выводимой при отсутствии устройства PROFINET IO.

При отказе станции PROFINET IO после деактивации через D_ACT_DP, операционная система не распознает отказ.

Активация устройства PROFINET IO

После повторной активации устройства PROFINET IO через D_ACT_DP, соответствующий контроллер PROFINET IO конфигурирует и параметрирует компонент (как при восстановлении неисправной станции PROFINET IO). Такая активация считается завершенной, если компонент может передавать данные пользователя.

При попытке активации с помощью инструкции D_ACT_DP устройства PROFINET IO, доступ к которому невозможен (напр., если оно физически отсоединено от шины), инструкция по истечении сконфигурированного времени параметрировния для распределенной периферии возвращает код ошибки W#16#80A7. Устройство PROFINET IO активируется, а тот факт, что доступ к активированному устройству PROFINET IO невозможен, приводит к соответствующей индикации в области системной диагностики.

Если после доступ к устройству PROFINET IO снова станет возможным, то система переходит в штатный режим.

Примечание

Активация устройства PROFINET IO может занять много времени. Для отмены выполняющегося задание активации, следует запустить D_ACT_DP с тем же значением для LADDR и MODE = 2. Повторять вызов D_ACT_DP с MODE = 2 до тех пор, пока не будет показана успешная отмена задания активации через RET_VAL = 0.

Параметр

В приведенной ниже таблице показаны параметры инструкции D_ACT_DP:

Параметр	Объявление	Тип данных	Описание
REQ	IN	Bool	Запускаемый по уровню управляющий параметр REQ = 1: Выполнить активацию или деактивацию
MODE	IN	USInt	Идентификатор задания Возможные значения: • 0: Запрос информации о том, активирован или деактивирован адресованный компонент (вывод с помощью параметра RET_VAL) • 1: Активация устройства PROFINET IO • 2: Деактивация устройства PROFINET IO
LADDR	IN	HW_DEVICE	Аппаратный идентификатор устройства PROFINET IO (HW_Device) Номер можно узнать в свойствах устройства PROFINET IO на вкладке сети или на вкладке "Системные константы" в стандартной таблице переменных (тегов). Если там указан как идентификатор для диагностики устройства, так и ID для смены рабочего состояния, то следует использовать код для диагностики устройства.
RET_VAL	OUT	Int	Если во время обработки функции возникает ошибка, то в возвращаемом значении будет код ошибки.
BUSY	OUT	Bool	Активный код: • BUSY = 1: Задание еще активно. • BUSY = 0: Задание завершено.

Параметр RET_VAL

Код ошибки* (W#16#...)	Значение
0000	Задание выполнено без ошибок.
0001	Устройство PROFINET IO активно (этот код ошибки возможен только при MODE = 0)
0002	Устройство PROFINET IO деактивировано (этот код ошибки возможен только при MODE = 0)
7000	Первый вызов с REQ = 0: Заданное с помощью LADDR задание неактивно; BUSY имеет значение "0".
7001	Первый вызов с REQ = 1: Программа запустила указанное в LADDR задание. BUSY имеет значение "1".
7002	Промежуточный вызов (REQ не релевантно). Активное задание еще активно; BUSY имеет значение 1.
8090	• Модуль с указанным в LADDR адресом не сконфигурирован. • Модуль CPU используется как I-Slave/I-Device и адрес этого I-Slave/I-Device был указан в LADDR.

Код ошибки* (W#16#...)	**Значение**
8092	Деактивация текущего адресуемого устройства PROFINET IO (MODE = 2) не может быть отменена активацией (MODE = 1). Активировать компонент позже.
8093	Указанный в LADDR адрес не относится к устройству PROFINET IO, которое может быть активировано или деактивировано, или параметр MODE неизвестен.
8094	Попытка активировать устройство, которое является потенциальным партнером для порта смены инструмента. Но в это время на этом порту смены инструмента уже активировано другое устройство. Активированное устройство остается активированным.
80A0	Ошибка связи между CPU и контроллером ввода/вывода.
80A1	Невозможно параметрировать адресованный компонент. (Этот код ошибки возможен только при MODE = 1.) Примечание: Если этот компонент при параметрировании активного устройства снова выйдет из строя, то инструкция D_ACT_DP вернет информацию об ошибке. Если параметрирование отдельного модуля не выполнено, то D_ACT_DP выводит информацию об ошибке W#16#0000.
80A3	Затронутый контроллер PROFINET IO не поддерживает эту функцию.
80A4	Модуль CPU не поддерживает эту функцию для внешнего контроллера PROFINET IO.
80A6	Ошибка слота в устройстве PROFINET IO; доступ возможен не ко всем данным пользователя (этот код ошибки возможен только при MODE =1). Примечание: D_ACT_DP выводит эту информацию об ошибке только в том случае, если происходит повторный отказ активированного компонента после параметрирования и до завершения выполнения инструкции D_ACT_DP. Если недоступен только один единственный модуль, то D_ACT_DP выводит информацию об ошибке W#16#0000.
80A7	При активации произошло превышение времени: Удаленное устройство недоступно или было установлено слишком короткое время параметрирования для центральной и распределенной периферии. Состояние удаленного устройства "активировано", но доступ невозможен.
80AA	Активация с ошибкой в устройстве PROFINET IO: Различия в конфигурации
80AB	Активация с ошибкой в устройстве PROFINET IO: Ошибка параметрирования
80AC	Активация с ошибкой в устройстве PROFINET IO: Требуется техобслуживание
80C1	D_ACT_DP была запущена и продолжается с другого адреса (этот код ошибки возможен при MODE =1 и при MODE =2).
80C3	• Временная ошибка ресурсов: В настоящий момент CPU обрабатывается максимальное число заданий активации и деактивации (8). (Этот код ошибки возможен, только если MODE = 1 и MODE= 2.) • CPU в настоящий момент принимает измененную конфигурацию. Активация/деактивация устройства PROFINET IO в настоящий момент невозможна.
80C6	PROFINET: Задания, не зарегистрированные пользователем, сбрасываются при перезапуске.
Общая информация об ошибках	Информацию для доступа к ошибке можно найти в описании инструкции GET_ERROR_ID (Страница 320).
* Коды ошибок могут отображаться в редакторе текстов программ в виде целых или шестнадцатеричных значений.	

9.3.9 Параметр STATUS для RDREC, WRREC и RALRM

Выходной параметр STATUS содержит информация об ошибке, которая обрабатывается как ARRAY[1...4] OF BYTE со следующей структурой:

Таблица 9-85 Выходной массив STATUS

Элемент массива	Название	Описание
STATUS[1]	Function_Num	• B#16#00, если не ошибок • ID функции DPV1-PDU: Если возникает ошибка , B#16#80 связывается через логическую функцию ИЛИ (при чтении набора данных с: B#16#DE; при записи набора данных с: B#16#DF). Если элемент протокола DPV1 не используется, то выводится B#16#C0 .
STATUS[2]	Error_Decode	Адрес ID ошибки
STATUS[3]	Error_Code_1	ID ошибки
STATUS[4]	Error_Code_2	Специфическое расширение идентификатора ошибки от производителя

Таблица 9-86 Значения STATUS[2]

Error_decode (B#16#....)	Источник	Описание
От 00 до 7F	CPU	Ошибок нет, предупреждений нет
80	DPV1	Ошибка согласно IEC 61158-6
От 81 до 8F	CPU	B#16#8x показывает ошибку в определенном параметре вызова инструкции.
FE, FF	Профиль DP	Специфическая ошибка профиля

Таблица 9-87 Значения STATUS[3]

Error_decode (B#16#....)	Error_code_1 (B#16#....)	Объяснение (DVP1)	Описание
00	00		Ошибок нет, предупреждений нет
70	00	Зарезервировано, отклонить	Первый вызов; нет активной передачи набора данных
	01	Зарезервировано, отклонить	Первый вызов; передача набора данных началась
	02	Зарезервировано, отклонить	Промежуточный вызов; передача набора данных уже активна

Error_decode (B#16#....)	Error_code_1 (B#16#....)	Объяснение (DVP1)	Описание
80	90	Зарезервирован, передать	Неправильный логический начальный адрес
	92	Зарезервирован, передать	Недопустимый тип для указателя Variant
	93	Зарезервирован, передать	Адресуемый с помощью ID или F_ID компонент DP не сконфигурирован.
	96		Инструкция RALRM (Страница 395) не может предоставить пусковую информацию, административную информацию, информацию заголовка или дополнительную информацию о прерывании OB. Для следующих OB можно использовать инструкцию DPNRM_DG (Страница 417) для асинхронного чтения актуальных диагностических сообщений затронутого ведомого устройства DP (информация об адресе из пусковой информации OB): • Аппаратное прерывание (Страница 94) • Состояние (Страница 100), обновление (Страница 101) или профиль (Страница 101) • Диагностическое прерывание (Страница 96) • Удаление или вставка модулей (Страница 98)
	A0	Ошибка чтения	Отрицательное подтверждение при чтении из модуля
	A1	Ошибка записи	Отрицательное подтверждение при записи в модуль
	A2	Отказ модуля	Ошибка протокола DP на уровне 2 (пример: отказ ведомого устройства или проблемы с шиной)
	A3	Зарезервирован, передать	• PROFIBUS DP: Ошибка протокола DP в Direct-Data-Link-Mapper или в User-Interface/User • PROFINET IO: Общая ошибка CM
	A4	Зарезервирован, передать	Нарушение связи по коммуникационной шине
	A5	Зарезервирован, передать	-
	A7	Зарезервирован, передать	Ведомое устройство DP или модули заняты (временная ошибка).
	A8	Конфликт версий	Ведомое устройство DP или модуль сообщает о несовместимости версий.
	A9	Функция не поддерживается	Функция не поддерживается ведомым устройством DP или модулем
	От AA до AF	Для конкретного пользователя	Ведомое устройство DP или модуль сообщает об определенной изготовителем ошибке в приложении. Обратиться к документации изготовителя ведомого устройства DP или модуля.
	B0	Неправильный индекс	Набор данных в модуле неизвестен; неправильный номер набора данных ≥ 256
	B1	Ошибка длины при записи	Неправильная информация о длине в параметре RECORD . • Для RALRM: Ошибка длины в AINFO

Error_decode (B#16#....)	Error_code_1 (B#16#....)	Объяснение (DVP1)	Описание
			Примечание: Интерактивная информационная система STEP 7 предлагает прямой доступ к информации по анализу возвращаемого AINFO буфера. • Для RDREC (Страница 387) и WRREC (Страница 387): Ошибка длины в MLEN
	B2	Неправильный слот	Сконфигурированный слот не занят.
	B3	Несоответствие типов	Фактический тип модуля не соответствует указанному типу модуля.
	B4	Недействительная область	Ведомое устройство DP или модуль сообщает о доступе к недействительной области данных.
	B5	Конфликт состояний	Ведомое устройство DP или модуль не готовы к работе
	B6	Доступ запрещен	Ведомое устройство DP или модуль отклоняют доступ.
	B7	Недействительный диапазон	Ведомое устройство DP или модуль сообщают о недействительном диапазоне для параметра или значения.
	B8	Недопустимый параметр	Ведомое устройство DP или модуль сообщают о неправильном параметре.
	B9	Недействительный тип	Ведомое устройство DP или модуль сообщают о неправильном типе: • Для RDREC (Страница 387): Буфер слишком мал (подмножества не могут быть прочитаны) • Для WRREC (Страница 387): Буфер слишком мал (подмножества не могут быть записаны)
	От BA до BF	Для конкретного пользователя	Ведомое устройство DP или модуль сообщает об определенной изготовителем ошибке доступа. Обратиться к документации изготовителя ведомого устройства DP или модуля.
	C0	Конфликт ограничений чтения	• Для WRREC (Страница 387): Запись данных возможна только для CPU в режиме работы STOP. **Примечание**: Это означает, что данные не могут записываться программой пользователя. Данные можно записывать только в режиме онлайн с помощью PG/PC. • Для RDREC (Страница 387): Модуль перенаправляет блок данных, но данные либо отсутствуют или чтение данных возможно только для CPU в режиме работы STOP. **Примечание**: Если запись данных возможно только в режиме работы STOP, то обработка через программу пользователя невозможна. Данные можно считывать только в режиме онлайн с помощью PG/PC.
	C1	Конфликт ограничений записи	Данные от предыдущего запроса на запись в модуль для того же набора данных еще не обработаны модулем.
	C2	Ресурс занят	В настоящее время модуль обрабатывает максимально возможное количество заданий для CPU.
	C3	Ресурсы отсутствуют	Необходимые ресурсы в настоящий момент заняты.

Error_decode (B#16#....)	Error_code_1 (B#16#....)	Объяснение (DVP1)	Описание
	C4		Внутренняя временная ошибка. Не удалось выполнить задание. Повторить задание. При частом повторении ошибки проверить установку на отсутствие источников электрических помех.
	C5		Ведомое устройство DP или модуль недоступны
	C6		Передача набора данных была отменена из-за конфликта классов приоритетов.
	C7		Отмена задания из-за перезапуска (теплый пуск) или холодного пуска ведущего устройства DP.
	От C8 до CF		Ведомое устройство DP или модуль сообщает об определенной изготовителем ошибке ресурсов. Обратиться к документации изготовителя ведомого устройства DP или модуля.
	DX	Для конкретного пользователя	Для конкретного ведомого устройства DP См. описание ведомого устройства DP.
81	От 00 до FF		Ошибка в первом параметре вызова (для RALRM (Страница 395): MODE)
	00		Недопустимый режим работы
82	От 00 до FF		Ошибка во втором параметре вызова
88	От 00 до FF		Ошибка в восьмом параметре вызова (для RALRM (Страница 395): TINFO) **Примечание**: Интерактивная информационная система STEP 7 предлагает прямой доступ к информации по анализу возвращаемого TINFO буфера.
	01		Неверный ID синтаксиса
	23		Превышение количественной структуры или слишком маленькая целевая область
	24		Неверный ID области
	32		Номер DB/DI вне области пользователя
	3A		Номер DB/DI равен НУЛЮ для ID области DB/DI или указанный DB/DI не существует.
89	От 00 до FF		Ошибка в девятом параметре вызова (для RALRM (Страница 395): AINFO) **Примечание**: Интерактивная информационная система STEP 7 предлагает прямой доступ к информации по анализу возвращаемого AINFO буфера.
	01		Неверный ID синтаксиса
	23		Превышение количественной структуры или слишком маленькая целевая область
	24		Неверный ID области
	32		Номер DB/DI вне области пользователя
	3A		Номер DB/DI равен НУЛЮ для ID области DB/DI или указанный DB/DI не существует.
8A	От 00 до FF		Ошибка в десятом параметре вызова
8F	От 00 до FF		Ошибка в пятнадцатом параметре вызова
FE, FF	От 00 до FF		Специфическая ошибка профиля

Элемент массива STATUS[4]

При ошибках DPV1 ведущее устройство DP перенаправляется STATUS[4] на CPU и в инструкцию. Без ошибки DPV1 это значение установлено на "0", со следующими исключениями для RDREC:

- STATUS[4] содержит длину целевой области RECORD, если MLEN > длина целевой области из RECORD.
- STATUS[4] = MLEN, если фактическая длина набора данных < MLEN < длина целевой области из RECORD.
- STATUS[4] = 0, если STATUS[4] > 255; требуется установка

В PROFINET IO значение STATUS[4] = 0.

9.3.10 Другие

9.3.10.1 DPRD_DAT und DPWR_DAT (чтение/запись непротиворечивых данных)

С помощью инструкции DPRD_DAT (чтение непротиворечивых данных) выполняется последовательное чтение одного или нескольких байт данных, а с помощью инструкции DPWR_DAT (запись непротиворечивых данных) выполняется последовательная передача одного или нескольких байт данных. Инструкция DPRD_DAT и DPWR_DAT может использоваться для PROFINET и PROFIBUS.

Таблица 9-88 Инструкции DPRD_DAT и DPWR_DAT

LAD/FBD	SCL	Описание

	`ret_val := DPRD_DAT(`
	`  laddr:=_word_in_,`
	`  record=>_variant_out_);`
		С помощью инструкции DPRD_DAT выполняется чтение одного или нескольких байт данных из модулей или субмодулей в одном из следующих мест хранения:
		• Локальные базовые I/O
		• Ведомое устройство DP
		• Устройство PROFINET IO
		CPU последовательно передает считанные данные. Если при передаче данных не возникли ошибки, то считанные данные записываются CPU в целевую область, указанную в параметре RECORD. Длина целевой области должна совпадать с длиной, сконфигурированной для выбранного модуля в STEP 7. При выполнении инструкции DPRD_DAT возможно обращение только к данным одного модуля или субмодуля соответственно. Передача начинается со сконфигурированного начального адреса.

	`ret_val := DPWR_DAT(`
	`  laddr:=_word_in_,`
	`  record:=_variant_in_);`
		Использовать инструкцию DPWR_DAT для последовательной передачи данных из RECORD в следующие места хранения:
		• Адресованный модуль или субмодуль в локальном базовом устройстве
		• Стандартное ведомое устройство DP
		• Устройство PROFINET IO
		Длина целевой области должна совпадать с длиной, сконфигурированной для выбранного модуля или субмодуля в STEP 7.

- S7-1200 CPU поддерживает последовательное считывание или запись 1, 2 или 4 байт в периферийных устойствах. С помощью инструкции DPRD_DAT выполняется последовательное считывание данных с отличной от 1, 2 или 4 байт длиной, а с помощью инструкции DPWR_DAT выполняется последовательная запись таких данных.
- Эти инструкции могут использоваться для областей данных в 1 байт или несколько байт. При отказе в доступе сообщается код ошибки W#16#8090.
- PROFINET поддерживает до 1024 байт непротиворечивых данных. Для последовательной передачи между S7-1200 и устройствами PROFINET эти инструкции не требуются.

Примечание

При использовании инструкций DPRD_DAT и DPWR_DAT с непротиворечивыми данными следует исключить эти непротиворечивые данные из автоматического обновления образа процесса. Дополнительную информацию можно найти в "Основы PLC: Выполнение программы пользователя" (Страница 83).

Таблица 9-89 Параметр

Параметр	Объявление	Тип данных	Описание
LADDR	IN	HW_IO (Word)	ID оборудования для модуля, из которого должны считываться данные. (DPRD_DAT). ID оборудования для модуля, в который должны записываться данные (DPWR_DAT). ID оборудования можно узнать из характеристик модуля на вкладке оборудования или в системных константах.
RECORD	OUT	Variant	Целевая область считанных данных пользователя (DPRD_DAT) или исходная область записываемых данных пользователя (DPWR_DAT). Размер этой области должен совпадать с размером области, сконфигурированным в выбранном модуле с помощью STEP 7.
RET_VAL	OUT	Int	Если во время работы функции возникает ошибка, то в возвращаемом значении будет код ошибки.

Принцип работы DPRD_DAT

С помощью параметра LADDR выбирается модуль стандартного ведомого устройства DP/устройства PROFINET IO. При наличии ошибки доступа на адресованном модуле возвращается код ошибки W#16#8090.

С помощью параметра RECORD определяется целевая область считанных данных:

- Целевая область должна иметь как минимум такую же длину, что и входы выбранного модуля. Передаются только входы; остальные байты не учитываются. При чтении из стандартного ведомого устройства DP с модульной структурой или с несколькими идентификаторами DP, при каждом вызове инструкции DPRD_DAT возможно обращение только к данным одного модуля сконфигурированного идентификатора оборудования. При выборе слишком маленькой целевой области, в параметре RET_VAL выводится код ошибки W#16#80B1.
- Могут использоваться следующие типы данных: Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UDInt. Использование этих типов данных в структурах пользовательского типа данных (UDT) с типом ARRAY или STRUCT допускается.
- Тип данных STRING не поддерживается.
- Если при передаче данных не возникли ошибки, то считанные данные записываются в целевую область, указанную в параметре RECORD.

Принцип работы DPWR_DAT

С помощью параметра LADDR выбирается модуль стандартного ведомого устройства DP/устройства PROFINET IO. При наличии ошибки доступа на адресованном модуле возвращается код ошибки W#16#8090.

С помощью параметра RECORD определяется исходная область записываемых данных:

- Исходная область должна иметь как минимум такую же длину, что и выходы выбранного модуля. Передаются только выходы; остальные байты не учитываются. Если исходная область в параметре RECORD длиннее, чем выходы сконфигурированного модуля, то передаются только данные до максимальной длины выходов. Если исходная область в параметре RECORD короче, чем выходы сконфигурированного модуля, то в параметре RET_VAL возвращается код ошибки W#16#80B1.
- Могут использоваться следующие типы данных: Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UDInt. Использование этих типов данных в структурах пользовательского типа данных (UDT) с типом ARRAY или STRUCT допускается.
- Тип данных STRING не поддерживается.
- Данные передаются синхронно, т.е. процесс записи завершается вместе с инструкцией.

Коды ошибок

Таблица 9-90 Коды ошибок DPRD_DAT и DPWR_DAT

Код ошибки [1]	Описание
0000	Ошибки отсутствуют
8090	Возможна одна из следующих ситуаций: • Для указанного логического базового адреса не был сконфигурирован модуль. • Было проигнорировано ограничение касательно длины непротиворечивых данных. • Начальный адрес в параметре LADDR указан не в шестнадцатеричном формате.

Код ошибки [1]	Описание
8092	Параметр RECORD поддерживает следующие типы данных: Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UDInt, and arrays of these types.
8093	По указанному в LADDR логическому адресу нет модуля DP/устройства PROFINET IO, из которого могут быть считаны (DPRD_DAT) или в который могут быть записаны непротиворечивые данные (DPWR_DAT).
80A0	При доступе к периферии обнаружен ошибка доступа (DPRD_DAT).
80B1	Длина указанной целевой области (DPRD_DAT) или исходной области (DPWR_DAT) не совпадает со сконфигурированной в STEP 7 длиной данных пользователя.
80B2	Системная ошибка с внешним DP интерфейсным модулем (DPRD_DAT) и (DPWR_DAT)

[1] Для отображения кодов ошибок DPRD_DAT и DPWR_DAT следует использовать тип данных Word.

Примечание

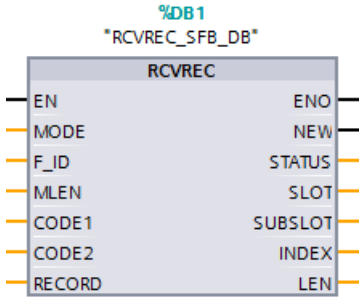
При обращении к ведомым устройствам DPV1, информация об ошибках этих ведомых устройств может перенаправляться ведущим устройством DP в инструкцию.

9.3.10.2 RCVREC (получение набора данных)

Интеллектуальное устройство (I-Device) может получать набор данных от вышестоящего контроллера. Пример выполняется в программе пользователя с помощью инструкции RCVREC (получение набора данных).

Таблица 9-91 Инструкция RCVREC

LAD/FBD	SCL	Описание

	`"RCVREC_SFB_DB"(`
	`mode:=_int_in_,`
	`F_ID:=_uint_in_,`
	`mlen:=_uint_in_,`
	`code1:=_byte_in_,`
	`code2:=_byte_in_,`
	`new=>_bool_out_,`
	`status=>_dword_out_,`
	`slot=>_uint_out_,`
	`subslot=>_uint_out_,`
	`index=>_uint_out_`
	`len=>_uint_out_,`
	`record:=_variant_inout_);`
		С помощью инструкции RCVREC можно получить набор данных от вышестоящего контроллера.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "RCVREC_SFB_DB" - это имя DB экземпляра.

Инструкция имеет следующие рабочие состояния:

- Проверка, есть ли у I-Device запрос на прием набора данных
- Предоставление набора данных выходным параметрам
- Отправка ответа в вышестоящий контроллер

Для определения обработанного инструкцией рабочего состояния можно использовать входной параметр MODE.

Интеллектуальное устройство должно находиться в рабочем состоянии RUN или STARTUP.

С помощью MLEN указывается максимальное количество байт для приема. Выбранная длина целевой области RECORD как минимум должна совпадать с длиной указанных MLEN байтов.

Если принимается набор данных (MODE = 1 или MODE = 2), то выходной параметр NEW сообщает, что набор данных сохранен в RECORD. Проследить, чтобы RECORD имела достаточную длину. Выходной параметр LEN содержит фактическую длину полученного набора данных в байтах.

Установить CODE1 иCODE2 на 0 для передачи положительного ответа на вышестоящий контроллер. Если полученный набор данных необходимо отклонить, то ввести отрицательный ответ для вышестоящего контроллера в коде ошибки 1 из CODE1 и в коде ошибки 2 из CODE2.

Примечание

Если интеллектуальное устройство получило запрос на получение набора данных, то в течение определенного времени необходимо подтвердить получении этого запроса. После подтверждения в течение этого промежутка времени должен быть отправлен ответ на вышестоящий контроллер. В ином случае на I-Device возникает ошибка по превышению времени, на что ОС интеллектуального устройства передает отрицательный ответ на вышестоящий контроллер. Информацию о значении для промежутка времени можно найти в Технических данных модуля CPU.

После возникновения ошибки выходной параметр STATUS получает информацию об ошибке.

Рабочие состояния

Для определения рабочего состояния инструкции RCVREC можно использовать входной параметр MODE. Это шаг поясняется в следующей таблице:

MODE	Значение
0	Проверка, есть ли запрос на прием набора данных Если в интеллектуальном устройстве имеется набор данных от вышестоящего контроллера, то инструкция выполняет запись только в выходные параметры NEW, SLOT, SUBSLOT, INDEX и LEN. Если инструкция вызывается несколько раз с MODE = 0, то выходной параметр всегда будет относится к одному и тому же запросу.
1	Получение набора данных для любого субслота интеллектуального устройства Если в интеллектуальном устройстве имеется набор данных от вышестоящего контроллера для любого субслота I-Device, то инструкция выполняет запись в выходные параметры и передает набор данных в параметр RECORD.
2	Получение набора данных для конкретного субслота интеллектуального устройства Если в интеллектуальном устройстве имеется набор данных от вышестоящего контроллера для конкретного субслота I-Device, то инструкция выполняет запись в выходные параметры и передает набор данных в параметр RECORD.

MODE	Значение
3	Отправка положительного ответа вышестоящему контроллеру Инструкция проверяет запрос вышестоящего контроллера на получение набора данных, подтверждает имеющийся набор данных и отправляет положительное подтверждение вышестоящему контроллеру.
4	Отправка отрицательного ответа вышестоящему контроллеру Инструкция проверяет запрос вышестоящего контроллера на получение набора данных, отклоняет имеющийся набор данных и отправляет отрицательное подтверждение вышестоящему контроллеру. Причина отклонения вводится во входные параметры CODE1 и CODE2.

Примечание

После получения набора данных (NEW = 1) инструкцию RCVREC необходимо вызвать два раза, чтобы гарантировать полную обработку. При этом необходимо соблюдать следующую последовательность:

- Первый вызов с MODE = 1 или MODE = 2
- Второй вызов с MODE = 3 или MODE = 4

Параметр

В приведенной ниже таблице показаны параметры инструкции RCVREC:

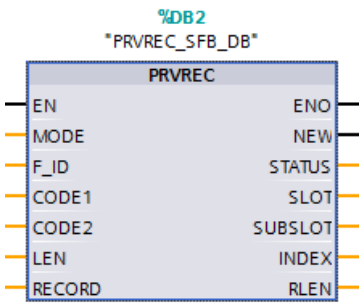
Параметр	Объявление	Тип данных	Описание
MODE	IN	Int	Режим работы
F_ID	IN	HW_SUBMODULE	Субслот в области передачи интеллектуального устройства для получаемого набора данных (имеет значение только для MODE = 2). Старшее слово всегда установлено на ноль.
MLEN	IN	Int	Максимальная длина получаемого набора данных в байтах.
CODE1	IN	Byte	Нуль (для MODE = 3) и/или код ошибки 1 (для MODE = 4)
CODE2	IN	Byte	Нуль (для MODE = 3) и/или код ошибки 2 (для MODE = 4)
NEW	OUT	Bool	• MODE = 0: Получен новый набор данных • MODE=1 или 2: Набор данных был передан в RECORD
STATUS	OUT	DWord	Информация об ошибке. Дополнительную информацию см. в "Параметр STATUS" (Страница 404).
SLOT	OUT	HW_SUBMODULE	Совпадает с F_ID
SUBSLOT	OUT	HW_SUBMODULE	Совпадает с F_ID
INDEX	OUT	UInt	Номер полученного набора данных
LEN	OUT	UInt	Длина полученного набора данных
RECORD	IN_OUT	Variant	Целевая область для полученного набора данных

9.3.10.3 PRVREC (предоставление набора данных)

Интеллектуальное устройство (I-Device) может получить запрос на представление набор данных от вышестоящего контроллера. Интеллектуальное устройство с помощью инструкции PRVREC (предоставление набора данных) открывает доступ к набору данных в программе пользователя.

Таблица 9-92 Инструкция PRVREC

LAD/FBD	SCL	Описание

	`"PRVREC_SFB_DB"(`
	`mode:=_int_in_,`
	`F_ID:=_uint_in_,`
	`code1:=_byte_in_,`
	`code2:=_byte_in_,`
	`len:=_uint_in_,`
	`new=>_bool_out_,`
	`status=>_dword_out_,`
	`slot=>_uint_out_,`
	`subslot=>_uint_out_,`
	`index=>_uint_out_`
	`rlen=>_uint_out_,`
	`record:=_variant_inout_);`
		С помощью инструкции PRVREC может быть получен запрос на представление набор данных от вышестоящего контроллера.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "PRVREC_SFB_DB" - это имя DB экземпляра.

Инструкция имеет следующие рабочие состояния:

- Проверка, есть ли у I-Device запрос на предоставление набора данных
- Передача запрошенного набора данных в вышестоящий контроллер
- Отправка ответа в вышестоящий контроллер

Для определения обработанного инструкцией рабочего состояния можно использовать входной параметр MODE.

Интеллектуальное устройство должно находиться в рабочем состоянии RUN или STARTUP.

Указать с помощью LEN макс. число байт, которое должен иметь передаваемый набор данных. Выбранная длина целевой области RECORD как минимум должна совпадать с длиной указанных LEN байтов.

Если имеется запрос на предоставление набора данных (MODE = 0), то выходной параметр NEW устанавливается на TRUE.

Если запрос на предоставление набора данных принимается, записать запрошенный набор данных в RECORD, чтобы отправить вышестоящему контроллеру положительный ответ, и записать значение 0 вCODE1 и CODE2. Если запрос на предоставление набора

данных необходимо отклонить, то ввести отрицательный ответ для вышестоящего контроллера в коде ошибки 1 из CODE1 и в коде ошибки 2 из CODE2.

Примечание

Если интеллектуальное устройство получило запрос на предоставление набора данных, то в течение определенного времени необходимо подтвердить получении этого запроса. После подтверждения в течение этого промежутка времени должен быть отправлен ответ на вышестоящий контроллер. В ином случае на I-Device возникает ошибка по превышению времени, на что ОС интеллектуального устройства передает отрицательный ответ на вышестоящий контроллер. Информацию о значении для промежутка времени можно найти в Технических данных модуля CPU.

После возникновения ошибки выходной параметр STATUS получает информацию об ошибке.

Рабочие состояния

Для определения рабочего состояния инструкции PRVREC можно использовать входной параметр MODE. Это шаг поясняется в следующей таблице:

MODE	Значение
0	Проверка, есть ли запрос на предоставление набора данных Если в интеллектуальном устройстве имеется запрос на предоставление набора данных от вышестоящего контроллера, то инструкция выполняет запись только в выходные параметры NEW, SLOT, SUBSLOT, INDEX и RLEN. Если инструкция вызывается несколько раз с MODE = 0, то выходной параметр всегда будет относится к одному и тому же запросу.
1	Получение запроса для предоставления набора данных для любого субслота интеллектуального устройства Если в интеллектуальном устройстве имеется такой запрос от вышестоящего контроллера для любого субслота I-Device, то инструкция выполняет запись в выходной параметр.
2	Получение запроса для предоставления набора данных для конкретного субслота интеллектуального устройства Если в интеллектуальном устройстве имеется такой запрос от вышестоящего контроллера для конкретного субслота I-Device, то инструкция выполняет запись в выходной параметр.
3	Предоставление блока данных и отправка положительного ответа вышестоящему контроллеру Инструкция проверяет запрос вышестоящего контроллера на предоставление набора данных, предоставляет набор данных в RECORD и отправляет положительное подтверждение вышестоящему контроллеру.
4	Отправка отрицательного ответа вышестоящему контроллеру Инструкция проверяет запрос вышестоящего контроллера на предоставление набора данных, отклоняет этот запрос и отправляет отрицательное подтверждение вышестоящему контроллеру. Причина отклонения вводится во входные параметры CODE1 и CODE2.

Примечание

После получения запроса (NEW = 1) инструкцию PRVREC необходимо вызвать два раза, чтобы гарантировать полную обработку. При этом необходимо соблюдать следующую последовательность:

- Первый вызов с MODE =1 или MODE = 2
- Второй вызов с MODE = 3 или MODE = 4

Параметр

В приведенной ниже таблице показаны параметры инструкции PRVREC:

Параметр	Объявление	Тип данных	Описание
MODE	IN	Int	Режим работы
F_ID	IN	HW_SUBMODULE	Субслот в области передачи интеллектуального устройства для передаваемого набора данных (имеет значение только для MODE = 2). Старшее слово всегда установлено на ноль.
CODE1	IN	Byte	Нуль (для MODE = 3) и/или код ошибки 1 (для MODE = 4)
CODE2	IN	Byte	Нуль (для MODE = 3) и/или код ошибки 2 (для MODE = 4)
LEN	IN	UInt	Максимальная длина передаваемого набора данных в байтах
NEW	OUT	Bool	Вышестоящий контроллер запросил новый набор данных
STATUS	OUT	DWord	Информация об ошибке. Дополнительную информацию см. в "Параметр STATUS" (Страница 404).
SLOT	OUT	HW_SUBMODULE	Совпадает с F_ID
SUBSLOT	OUT	HW_SUBMODULE	Совпадает с F_ID
INDEX	OUT	UInt	Номер передаваемого набора данных
RLEN	OUT	UInt	Длина передаваемого набора данных
RECORD	IN_OUT	Variant	Предоставленный набор данных

9.3.10.4 DPNRM_DG (чтение диагностических данных ведомого устройства DP)

Инструкция DPNRM_DG (чтение диагностических данных) может использоваться для PROFIBUS .

Таблица 9-93 Инструкция DPNRM_DG

LAD/FBD	SCL	Описание

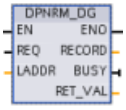
	`ret_val := DPNRM_DG(`
	`  req:=_bool_in_,`
	`  laddr:=_word_in_,`
	`  record=>_variant_out_,`
	`  busy=>_bool_out_);`
		С помощью инструкции DPNRM_DG могут быть считаны текущие диагностические данные ведомого устройства DP в формате, специфицированном в EN 50 170 Volume 2, PROFIBUS. Считанные данные после безошибочной передачи данных записываются в указанную RECORD целевую область.

Таблица 9-94 Типы данных для параметров DPNRM_DG

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	REQ=1: Запрос на чтение
LADDR	IN	HW_DPSLAVE	Сконфигурированный диагностический адрес ведомого устройства DP: это должен быть адрес станции, а не устройства ввода/вывода. Для определения диагностического адреса выбрать станцию (а не образ устройства) на вкладке сети окна конфигурации устройств. Ввести адрес в шестнадцатеричном формате. Пример: Диагностический адрес 1022 означает LADDR:=W#16#3FE.
RET_VAL	OUT	Int	Если во время работы функции возникает ошибка, то в возвращаемом значении будет код ошибки. Если ошибок не возникло, в RET_VAL вводится длина фактически переданных данных.
RECORD	OUT	Variant	Целевая область для считанных диагностических данных. Минимальная длина считываемого набора данных (или целевой области) составляет 6 байт. Максимальная длина передаваемого набора данных составляет 240 байт. Стандартные ведомые устройства могут передавать более 240 байт диагностических данных (максимум до 244 байт). Тогда только первые 240 байт передаются в целевую область, а в данных устанавливается бит переполнения.
BUSY	OUT	Bool	BUSY=1: Задание чтения еще не завершено.

Операция чтения запускается назначением значения 1 входному параметру REQ в вызове инструкции DPNRM_DG. Задание чтения выполняется асинхронно, другими словами, ему необходимы несколько вызовов инструкции DPNRM_DG. Состояние задания отображается в выходных параметрах RET_VAL и BUSY.

Таблица 9-95 Структура диагностических данных ведомого устройства

Байт	Описание
0	Состояние станции 1
1	Состояние станции 2
2	Состояние станции 3
3	Номер мастер-станции
4	ID изготовителя (старший байт)
5	ID изготовителя (младший байт)
6 ...	Дополнительная специфическая для ведомого устройства диагностическая информация

Таблица 9-96 Код ошибки инструкции DPNRM_DG

Код ошибки	Описание	Ограничение
0000	Ошибки отсутствуют	-
7000	Первый вызов с REQ = 0: Нет активной передачи данных; BUSY имеет значение 0.	-
7001	Первый вызов с REQ = 1: Нет активной передачи данных; BUSY имеет значение 1.	Распределенный I/O
7002	Промежуточный вызов (REQ не играет роли): Передача данных уже активна; BUSY имеет значение 1.	Распределенный I/O

Код ошибки	Описание	Ограничение
8090	Указанный логический базовый адрес недействителен: Базовый адрес отсутствует.	-
8092	Параметр RECORD поддерживает следующие типы данных: Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UDInt, and arrays of these types.	-
8093	• Эта инструкция не разрешена для указанного в LADDR модуля (модули S7-DP для S7-1200 разрешены). • LADDR указывает устройство ввода/вывода вместо станции. Для определения диагностического адреса для LADDR, выбрать станцию (а не образ устройства) на вкладке сети окна конфигурации устройств.	-
80A2	• Ошибка протокола DP на уровне 2 (пример: отказ ведомого устройства или проблемы с шиной) • Для ET200S набор данных не может быть считан в DPV0 режиме.	Распределенный I/O
80A3	Ошибка протокола DP с User-Interface/User	Распределенный I/O
80A4	Проблемы со связью по коммуникационной шине	Ошибка возникает между CPU и внешним интерфейсным DP-модулем.
80B0	• Инструкция не разрешена для типа модуля. • Модуль не распознает набор данных. • Номер набора данных 241 недопустим.	-
80B1	Указанная в параметре RECORD длина неправильная.	Указанная длина > Длина набора данных
80B2	Сконфигурированный слот не занят.	-
80B3	Фактический тип модуля не соответствует требуемому типу модуля.	-
80C0	Диагностическая информация отсутствует.	-
80C1	Данные от предыдущего запроса на запись в модуль для того же набора данных еще не обработаны модулем.	-
80C2	В настоящее время модуль обрабатывает максимально возможное количество заданий для CPU.	-
80C3	Необходимые ресурсы (память и т.д.) в настоящий момент заняты.	-
80C4	Внутренняя временная ошибка. Не удалось обработать задание. Повторить задание. При частом повторении ошибки, проверить систему на отсутствие источников электрических помех.	-
80C5	Распределенный I/O недоступен	Распределенный I/O
80C6	Передача набора данных была отменена из-за конфликта классов приоритетов (перезапуск или фоновый режим).	Распределенный I/O
8xyy[1]	Общие коды ошибок	

Дополнительную информацию об общих кодах ошибок можно найти в "Расширенные инструкции, распределенная периферия: информация об ошибках для RDREC, WRREC и RALRM" (Страница 404).

9.4 PROFIenergy

PROFIenergy - это не зависящий от производителя и устройства профиль для управления энергопотреблением с помощью PROFINET. Чтобы снизить потребление электроэнергии во время остановок производства и незапланированных простоев, можно отключать оборудование скоординированным и централизованным способом, используя PROFIenergy.

Контроллер PROFINET IO отключает устройства / силовые модули PROFINET с помощью специальных команд в программе пользователя. Дополнительное оборудование не требуется. Устройства PROFINET интерпретируют команды PROFIenergy напрямую.

S7-1200 CPU не поддерживает функциональность PE контроллеров. S7-1200 CPU может действовать только как объект PROFIenergy (с функциональностью интеллектуального устройства).

Контроллер PROFIenergy (PE контроллер)

PE контроллер - это высокоуровневый CPU (например, S7-1500), который активирует или деактивирует состояние покоя устройств нижнего уровня. PE контроллер деактивирует и повторно активирует определенные производственные компоненты или целые производственные линии с помощью программы пользователя. Устройства более низкого уровня получают команды от программы пользователя через соответствующие инструкции (функциональные блоки).

Программа пользователя отправляет команды, используя протокол коммуникации PROFINET. Команда PE может быть либо управляющей командой для перевода объекта PE в режим энергосбережения, либо командой для считывания состояния или измеренного значения.

С помощью инструкции PE_I_DEV данные запрашиваются из модуля. Программа пользователя должна определить, какая информация запрашивается PE контроллером, и получить ее из энергетического модуля, используя наборы данных. Сам модуль не поддерживает напрямую команды PE. Модуль сохраняет информацию об измерении энергии в общей области, а CPU нижнего уровня (например, S7-1200) запускает инструкцию PE_I_DEV , чтобы вывести информацию на PE контроллер.

PROFIenergy объект (PE объект)

PE объект (например, S7-1200) принимает команды PROFIenergy PE контроллера (например, S7-1500) и выполняет их соответствующим образом (например, путем возврата измеренного значения или активации режима энергосбережения). Реализация PE объекта в устройстве с поддержкой PROFIenergy зависит от устройства и изготовителя.

Справочная информация

Дополнительную информацию о PROFIenergy можно найти в системе интерактивной помощи STEP 7 в TIA Portal. Примеры использования инструкций PROFIenergy можно найти в Industry Online Support в статье "PROFIenergy - экономия энергии с SIMATIC S7 (http://support.automation.siemens.com/WW/view/en/41986454)".

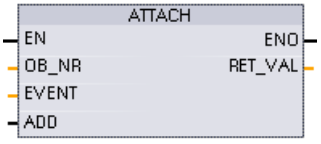
9.5 Прерывания

9.5.1 Инструкции ATTACH и DETACH (согласование OB и прерывающего события/отмена согласования)

С помощью инструкций ATTACH и DETACH можно активировать и деактивировать подпрограммы, запускаемые прерывающими событиями.

Таблица 9-97 Инструкции ATTACH и DETACH

LAD/FBD	SCL	Описание

	`ret_val := ATTACH(`
	`ob_nr:=_int_in_,`
	`event:=_event_att_in_,`
	`add:=_bool_in_);`
		Инструкция ATTACH активирует выполнение подпрограммы OB прерывания при аппаратном событии.

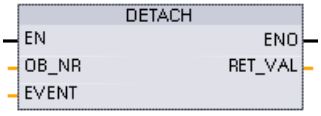
	`ret_val := DETACH(`
	`ob_nr:=_int_in_,`
	`event:=_event_att_ in);`
		Инструкция DETACH деактивирует выполнение подпрограммы OB прерывания при аппаратном событии.

Таблица 9-98 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
OB_NR	IN	OB_ATT	Идентификатор организационного блока: Выбрать один из доступных OB аппаратных прерываний, которые были созданы с помощью функции "Добавить новый блок". Выполнить двойной клик по полю параметра и затем кликнуть по символу, чтобы отобразить доступные OB.
EVENT	IN	EVENT_ATT	Идентификатор события: Выбрать одно из доступных событий аппаратного прерывания, которые были включены в конфигурации устройства PLC для цифровых входов или высокоскоростных счётчиков. Выполнить двойной клик по полю параметра и затем кликнуть по символу, чтобы отобразить доступные события.
ADD (толькоATTACH)	IN	Bool	• ADD = 0 (по умолчанию): Данное событие заменяет все имеющиеся на текущий момент назначенные события для этого OB. • ADD = 1: Данное событие добавляется ко всем имеющимся на текущий момент назначенным событиям для этого OB.
RET_VAL	OUT	Int	Условие выполнения

События аппаратных прерываний

Следующие события аппаратных прерываний поддерживаются CPU:

- События переднего фронта: первые 12 встроенных цифровых входов CPU (от DEa.0 до DEb.3) и все цифровые входы SB
  - Передний фронт имеет место тогда, когда цифровой вход переходит из ВЫКЛ на ВКЛ в ответ на изменение сигнала устройства, подключенного к входу.
- События заднего фронта: первые 12 встроенных цифровых входов CPU (от DEa.0 до DEb.3) и все цифровые входы SB
  - Задний фронт имеет место тогда, когда цифровой вход переходит из ВКЛ на ВЫКЛ.
- События: Текущее значение высокоскоростного счётчика (HSC) = Опорное значение (CV = RV) (HSC от 1 до 6)
  - Прерывание CV = RV генерируется для HSC тогда, когда текущее значение счетчика переходит от предыдущего значения к значению, которое точно соответствует установленному ранее опорному значению.
- События: Изменение направления счета HSC (HSC от 1 до 6)
  - Событие изменения направления счета происходит тогда, когда обнаруживается, что HSC переходит от прямого счета к обратному или от обратного к прямому.
- События: Внешний сброс HSC (HSC от 1 до 6)
  - Определенные режимы работы HSC допускают назначение цифрового входа для внешнего сброса значения счетчика HSC на ноль. Событие внешнего сброса происходит для такого HSC тогда, когда этот вход переходит из состояния ВЫКЛ в состояние ВКЛ.

Активация событий аппаратных прерываний в конфигурации устройства

Аппаратные прерывания должны быть активированы во время настройки устройства. Необходимо установить флажок для активации событий в конфигурации устройства для канала цифрового ввода или HSC, если потребуется назначить это событие во время настройки или при работе.

Возможные флажки-опции в конфигурации устройства PLC:

- Цифровой вход
  - Активация распознавания переднего фронта
  - Активация распознавания заднего фронта
- Высокоскоростной счётчик (HSC)
  - Активация этого высокоскоростного счётчика
  - Генерирование прерывания при равенстве текущего значения счетчика и опорного значения
  - Генерирование прерывания для события внешнего сброса
  - Генерирование прерывания для события направления счета

Вставка новых OB аппаратных прерываний в программу

По умолчанию при первой активации события OB ему не назначается. Для этого используется идентификатор "<не подключено>" в конфигурации устройства "Аппаратное прерывание:". Только OB аппаратных прерываний могут назначаться событию аппаратного прерывания. Все имеющиеся OB аппаратных прерываний перечислены в выпадающем списке "Аппаратное прерывание:". Если OB отсутствуют, то следует создать OB с типом "аппаратное прерывание" следующим образом. Выполнить следующие действия в ветке дерева проекта "Программные блоки":

1. Дважды кликнуть на "Добавление нового блока", выбрать вариант "Организационный блок (OB)" и после "Аппаратное прерывание".
2. Как опция можно переименовать OB, выбрать язык программирования (LAD, FBD или SCL) и номер блока (перейти на ручное редактирование и выбрать отличный от предложенного номер блока).
3. Отредактировать OB и вставить реакцию программы, которая должна последовать при наступлении события. Из этого OB могут вызывать FC и FB, вплоть до максимальной глубины вложенности. Максимальная глубина вложенности для программ обеспечения безопасности составляет четыре уровня. Для других программ максимальная глубина вложенности составляет шесть уровней.

Параметр OB_NR

Имена всех имеющиеся OB аппаратных прерываний перечисляются в выпадающем списке "Аппаратное прерывание:" в конфигурации устройства и в выпадающем списке для параметра "OB_NR" инструкций "ATTACH /DETACH".

Параметр EVENT

При активации события аппаратного прерывания ему присваивается однозначное имя события по умолчанию. Это имя события может быть изменено в поле редактирования "Имя события:", при этом имя должно оставаться однозначным. Имена событий преобразуются в имена переменных (тегов) в таблица переменных (тегов) "Константы" и появляются в выпадающем списке параметра "EVENT" в полях инструкций ATTACH иDETACH. Значение этих переменных (тегов) это внутренний номер, обозначающий событие.

Общий принцип действия

Каждое аппаратное прерывание может быть назначено OB аппаратного прерывания, который будет поставлен в очередь на выполнение при возникновении события аппаратного прерывания. Согласование OB и события может быть выполнено во время настройки и при работе.

При настройке можно назначить OB для активного события или отменить назначение. Для того, чтобы назначить OB событию во время настройки, выбрать OB в выпадающем списке доступных OB аппаратных прерываний в "Аппаратном прерывании:" (нажать на направленную вниз стрелку на выпадающем списке). Выбрать в этом списке название требуемого OB или выбрать "<не подключено>", чтобы отменить назначение.

Назначить и снять назначение для активированного события аппаратного прерывания можно и при выполнении программы. С помощью инструкций ATTACH и DETACH

соответствующему OB при выполнении программы (при необходимости и повторно) можно назначить событие аппаратного прерывания или отменить такое назначение. Если в настоящий момент нет назначенного OB (либо из-за выбора "<не подключено>" в конфигурации устройства, либо как результат выполнения DETACH), то активированное аппаратное прерывание игнорируется.

Принцип работы DETACH

С помощью инструкции DETACH отменяется назначение определенного события или событий определенному OB. Если EVENT указан, то только это одно событие отключается от указанного организационного блока OB_NR. Все другие назначенные этому организационному блоку OB_NR в настоящий момент события остаются подключенными. Если EVENT не указан, то все назначенные этому организационному блоку OB_NR в настоящий момент события отключаются.

Коды условий

Таблица 9-99 Коды условий

RET_VAL (W#16#....)	ENO	Описание
0000	1	Ошибки отсутствуют
0001	1	Назначение отсутствует (только DETACH)
8090	0	OB отсутствует
8091	0	OB относится к неправильному типу
8093	0	Событие отсутствует

9.5.2 Циклические прерывания

9.5.2.1 SET_CINT (параметрирование циклического прерывания)

Таблица 9-100 SET_CINT (параметрирование циклического прерывания)

LAD/FBD	SCL	Описание

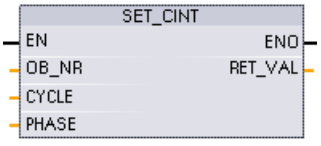
	`ret_val := SET_CINT(`
	`  ob_nr:=_int_in_,`
	`  cycle:=_udint_in_,`
	`  phase:=_udint_in_);`
		Указанный OB прерывания устанавливается для запуска циклического выполнения, прерывающего программный цикл.

Таблица 9-101 Типы данных для параметров

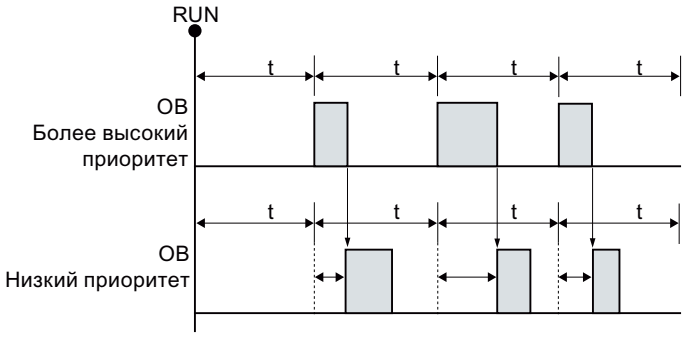
Параметр и тип данных		Тип данных	Описание
OB_NR	IN	OB_CYCLIC	Номер OB (только символьная адресация)
CYCLE	IN	UDInt	Интервал времени в микросекундах
PHASE	IN	UDInt	Фазовый сдвиг в микросекундах
RET_VAL	OUT	Int	Условие выполнения

Примеры: Параметры времени

- Если время CYCLE = 100 мкс, то OB прерывания, указанный в OB_NR, прерывает программный цикл каждые 100 мкс. OB прерывания выполняется и после снова передает управление выполнением программному циклу в том месте, где возникло прерывание.
- Если время CYCLE = 0, то прерывающее событие деактивируется и OB прерывания не выполняется.
- Время PHASE (фазовый сдвиг) это указанное время задержки до начала интервала времени CYCLE. С помощью фазового сдвига можно по времени управлять выполнением OB с более низким приоритетом.

При вызове OB с более низким и более высоким приоритетом в одном и том же интервале времени, OB с более низким приоритетом вызывается только после завершения обработки OB с более высоким приоритетом. Начало выполнения OB с более низким приоритетом может быть отложено, в зависимости от времени обработки OB с более высоким приоритетом.

Вызов OB без фазового сдвига



Если необходимо запустить выполнение OB с более низким приоритетом с фиксированным временем цикла, то фазовый сдвиг должен быть больше, чем время обработки OB с более высоким приоритетом.

Вызов OB с фазовым сдвигом

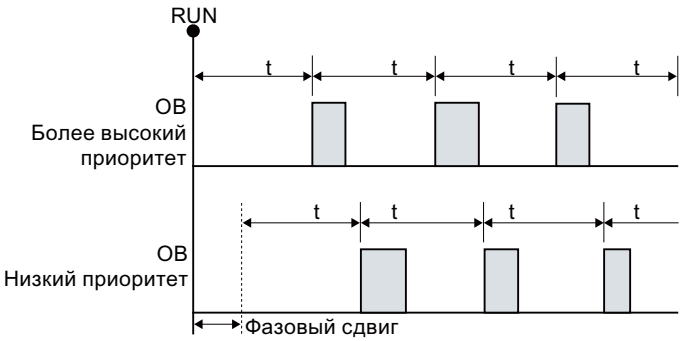


Таблица 9-102 Коды условий

RET_VAL (W#16#....)	Описание
0000	Ошибки отсутствуют
8090	OB отсутствует или имеет неверный тип
8091	Недопустимое время цикла
8092	Недействительное время фазового сдвига
80B2	У OB нет соответствующего события

9.5.2.2 QRY_CINT (запрос параметров циклического прерывания)

Таблица 9-103 **QRY_CINT (запрос циклического прерывания)**

LAD/FBD	SCL	Описание

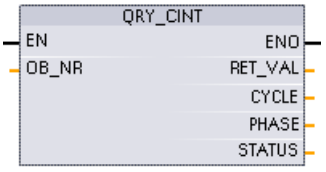
	`ret_val := QRY_CINT(`
	`ob_nr:=_int_in_,`
	`cycle=>_udint_out_,`
	`phase=>_udint_out__,`
	`status=>_word_out_);`
		Выполняется запрос параметров и состояния выполнения OB циклического прерывания. Возвращенные значения существуют на момент выполнения QRY_CINT.

Таблица 9-104 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
OB_NR	IN	OB_CYCLIC	Номер OB (символьная адресация, напр., OB_MyOBName)
RET_VAL	OUT	Int	Условие выполнения
CYCLE	OUT	UDInt	Интервал времени в микросекундах

Параметр и тип данных		Тип данных	Описание
PHASE	OUT	UDInt	Фазовый сдвиг в микросекундах
STATUS	OUT	Word	Код состояния циклического прерывания: • Биты от 0 до 4, см. таблицу STATUS ниже • Прочие биты, всегда 0

Таблица 9-105 Параметр STATUS

Бит	Значение	Описание
0	0	При CPU RUN
	1	При запуске
1	0	Прерывание активировано.
	1	Прерывание деактивируется через инструкцию DIS_IRT.
2	0	Прерывание не активно или истекло.
	1	Прерывание активно.
4	0	Указанный в OB_NR блок OB не существует.
	1	Указанный в OB_NR блок OB существует.
Прочие биты		Всегда 0

При возникновении ошибки, RET_VAL отображает соответствующий код ошибки и параметр STATUS = 0.

Таблица 9-106 Параметр RET_VAL

RET_VAL (W#16#....)	Описание
0000	Ошибки отсутствуют
8090	OB отсутствует или имеет неверный тип.
80B2	У OB нет соответствующего события.

9.5.3 Прерывания по времени

⚠ ПРЕДУПРЕЖДЕНИЕ

Опасность несанкционированного доступа к сетям пользователя при синхронизации по протоколу NTP (Network Time Protocol)

Если злоумышленник получит доступ к сетям пользователя посредством синхронизации по протоколу сетевого времени (NTP), то он может начать управлять процессом, изменив системное время CPU. Такая передача контроля над процессом может привести к смерти, серьезным травмам или материальному ущербу.

Функция клиента NTP для S7-1200 CPU по умолчанию отключена, и, после ее включения, только сконфигурированные IP-адреса могут выступать в качестве сервера NTP. CPU отключает эту функцию по умолчанию, и необходимо настроить эту функцию, чтобы разрешить удаленную коррекцию системного времени CPU.

S7-1200 CPU поддерживает прерывания и операции по времени, которые зависят от точности системного времени CPU. Если настраивается NTP и разрешается синхронизацию времени с сервера, то следует убедиться, что сервер является доверенным источником. Невыполнение этого требования может привести к нарушениям в системе безопасности, которые могут позволить неизвестному пользователю частично контролировать процесс, переставив системное время CPU.

Для получения информации и рекомендаций по обеспечению безопасности см. Operational Guidelines for Industrial Security (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) на сайте поддержки Siemens.

9.5.3.1 SET_TINTL (установка прерывания по времени)

Таблица 9-107 SET_TINTL (установка даты и времени для прерывания по времени с типом данных DTL)

LAD/FBD	SCL	Описание

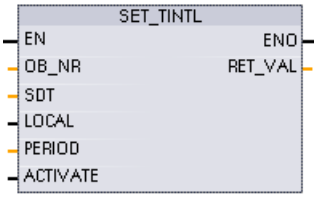
	`ret_val := SET_TINTL(`
	`OB_NR:=_int_in_,`
	`SDT:=_dtl_in_,`
	`LOCAL:=_bool_in_`
	`PERIOD:=_word_in_`
	`ACTIVATE:=_bool_in_);`
		Установка прерывания по времени. OB может быть установлен на однократное выполнение или для повторного выполнения в назначенный промежуток времени.

Таблица 9-108 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
OB_NR	IN	OB_TOD (INT)	Номер OB (только символьная адресация)
SDT	IN	DTL	Дата и время начала: секунды и миллисекунды игнорируются и могут быть установлены на 0).

Параметр и тип данных		Тип данных	Описание
LOCAL	IN	Bool	0 = Использовать системное время 1 = Использовать местное время (если для CPU установлено местное время, в остальных случаях следует использовать системное время)
PERIOD	IN	Word	Период времени от даты и времени начала для повторяющихся прерывающих событий. • W#16#0000 = однократно • W#16#0201 = каждую минуту • W#16#0401 = каждый час • W#16#1001 = ежедневно • W#16#1201 = еженедельно • W#16#1401 = ежемесячно • W#16#1801 = ежегодно • W#16#2001 = по окончании месяца
ACTIVATE	IN	Bool	0 = ACT_TINT должна быть выполнена для активации прерывающего события. 1 = Прерывающее событие активировано.
RET_VAL	OUT	Int	Условие выполнения

Программа с помощью инструкции SET_TINTL может определить дату и время прерывающего события, выполняющего назначенный OB прерывания. Дата и время запуска определяются параметром SDT, а промежуток времени для повторяющихся прерываний (напр., ежедневно или еженедельно) определяется параметром PERIOD. При установке повторяющегося ежемесячно времени, для даты запуска необходимо указать день от 1 до 28. Нельзя использовать дни от 29 до 31, так как они отсутствуют в феврале месяце. Если необходимо запрограммировать прерывающее событие по окончании месяца, то следует использовать для параметра PERIOD установку "конец месяца".

Значение дня недели DTL данных в параметре SDT игнорируется. Актуальная дата и актуальное время CPU устанавливаются с помощью функции "Установка времени" в окне "Онлайн и диагностика" Онлайн -CPU. Должны быть установлены день, месяц и год. STEP 7 рассчитывает период прерывания на основе даты и времени CPU.

Примечание

При переходе с летнего на зимнее время первый час дня отсутствует. Поэтому следует выбирать стартовое время со второго часа или использовать дополнительное прерывание по задержке времени в течение первого часа.

Таблица 9-109 Коды условий

RET_VAL (W#16#....)	Описание
0000	Ошибки отсутствуют
8090	Недопустимый параметр OB_NR

RET_VAL (W#16#....)	Описание
8091	Недопустимый параметр для начального времени SDT: (например, начальное время установлено на пропускаемый час в начале летнего времени)
8092	Недействительный параметр PERIOD
80A1	Начальное время в прошлом. (Этот код ошибки возможен только для PERIOD = W #16#0000.)

9.5.3.2 CAN_TINT (удаление прерывания по времени)

Таблица 9-110 CAN_TINT (удаление прерывания по времени)

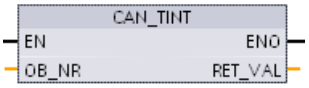
LAD/FBD	SCL	Описание
	`ret_val:=CAN_TINT(_int_in);`	Удаляет событие прерывания по времени для указанного OB прерывания.

Таблица 9-111 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
OB_NR	IN	OB_TOD (INT)	Номер OB (только символьная адресация)
RET_VAL	OUT	Int	Условие выполнения

Таблица 9-112 Коды условий

RET_VAL (W#16#....)	Описание
0000	Ошибки отсутствуют
8090	Недопустимый параметр OB_NR
80A0	Дата/время запуска для этого OB прерывания не установлено

9.5.3.3 ACT_TINT (активация прерывания по времени)

Таблица 9-113 ACT_TINT (активация прерывания по времени)

LAD/FBD	SCL	Описание
	`ret_val:=ACT_TINT(_int_in_);`	Активирует событие прерывания по времени для указанного OB прерывания.

Таблица 9-114 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
OB_NR	IN	OB_TOD (INT)	Номер OB (только символьная адресация)
RET_VAL	OUT	Int	Условие выполнения

Таблица 9-115 Коды условий

RET_VAL (W#16#....)	Описание
0000	Ошибки отсутствуют
8090	Недопустимый параметр OB_NR
80A0	Дата и время запуска не установлены для организационного блока прерывания по времени
80A1	Активированное время находится в прошлом. Ошибка появляется только в том случае, если OB прерывания настроен только на однократное выполнение.

9.5.3.4 QRY_TINT (запрос состояния прерывания по времени)

Таблица 9-116 QRY_TINT (запрос состояния прерывания по времени)

LAD/FBD	SCL	Описание

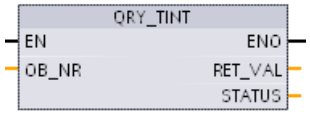
	`ret_val:=QRY_TINT(`
	`  OB_NR:=_int_in_,`
	`  STATUS=>_word_out_);`
		Запрашивает состояние прерывания по времени для указанного OB прерывания.

Таблица 9-117 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
OB_NR	IN	OB_TOD (INT)	Номер OB (только символьная адресация)
RET_VAL	OUT	Int	Условие выполнения
STATUS	OUT	Word	Состояние указанного OB прерывания

Таблица 9-118 Параметр STATUS

Бит	Значение	Описание
0	0	В режиме RUN
	1	При запуске

Бит	Значение	Описание
1	0	Прерывание активировано.
	1	Прерывание деактивировано.
2	0	Прерывание не активно или истекло.
	1	Прерывание активно.
4	0	Назначенный OB_NR отсутствует.
	1	OB с назначенным OB_NR существует.
6	1	Прерывание по времени использует местное время.
	0	Прерывание по времени использует системное время.
Прочее		Всегда 0

Таблица 9-119 Коды условий

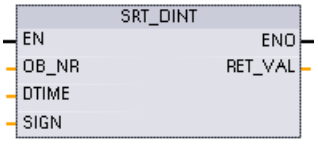
RET_VAL (W#16#....)	Описание
0000	Ошибки отсутствуют
8090	Недопустимый параметр OB_NR

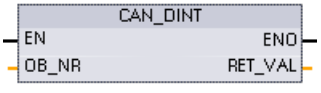
9.5.4 Прерывания по задержке времени

Можно запускать и отменять обработку прерывания по задержке времени с помощью инструкций SRT_DINT и CAN_DINT, а для запроса состояния прерывания использоваться инструкцию QRY_DINT. Каждое прерывание по задержке времени является однократным событием, которое наступает по истечении заданного времени задержки. Если событие задержки времени удаляется до истечения задержки времени, то прерывание не используется в программе.

Таблица 9-120 Инструкции SRT_DINT, CAN_DINT и QRY_DINT

LAD/FBD	SCL	Описание

	`ret_val := SRT_DINT(`
	`ob_nr:=_int_in_,`
	`dtime:=_time_in_,`
	`sign:=_word_in_);`
		SRT_DINT запускает прерывание по задержке времени, выполняющее OB, если установленное в параметре DTIME время задержки истекло.

	`ret_val := CAN_DINT(` `ob_nr:=_int_in_);`	CAN_DINT удаляет уже запущенное прерывание по задержке времени. В таком случае OB прерывания по задержке времени не выполняется.

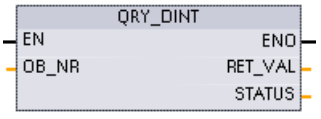
	`ret_val := QRY_DINT(`
	`ob_nr:=_int_in_,`
	
	`status=>_word_out_);`
		QRY_DINT запрашивает состояние указанного параметром OB_NR прерывания по задержке времени.

Таблица 9-121 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
OB_NR	IN	OB_DELAY	Организационный блок (OB), который должен быть запущен после задержки по времени: Выбрать один из доступных OB прерывания по задержке времени, которые были созданы с помощью функции "Добавить новый блок" в дереве проекта. Выполнить двойной клик по полю параметра и затем кликнуть по символу, чтобы отобразить доступные OB.
DTIME [1]	IN	Time	Значение задержки (от 1 до 60000 мс)
SIGN [1]	IN	Word	Не используется S7-1200: принимается любое значение. Для исключения ошибок необходимо присвоить значение.
RET_VAL	OUT	Int	Условие выполнения
STATUS	OUT	Word	Инструкция QRY_DINT: Состояние указанного OB прерывания по задержке времени см. таблицу ниже

[1] Только для SRT_DINT

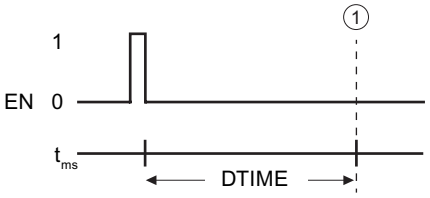
Выполнение

Если EN=1, то инструкция SRT_DINT запускает внутреннюю задержку по времени (DTIME). По истечении задержки времени, CPU генерирует прерывание программы, которое запускает выполнение OB прерывания по задержке времени. С помощью инструкции CAN_DINT можно аннулировать запущенное прерывание по задержке времени до наступления указанной задержки по времени. Всего может быть активно максимум четыре события прерывания по задержке времени.

Примечание

SRT_DINT запускает таймер для задержки по времени в каждом цикле, если EN = 1. Однократно подтвердить EN = 1 вместо установки EN = 1, чтобы запустить задержку по времени.

Временная диаграмма для инструкции SRT_DINT:



① Прерывание по задержке времени выполняется

Вставка OB прерывания по задержке времени в проект

Добавление OB прерывания по задержке времени возможно только в инструкциях SRT_DINT и CAN_DINT. В новом проекте еще нет OB прерывания по задержке времени. Необходимо вставить OB прерывания по задержке времени в проект. Для создания OB прерывания по задержке времени необходимо выполнить следующее:

1. Дважды кликнуть в дереве проекта в "Программных блоках" на "Добавление нового блока", выбрать вариант "Организационный блок (OB)" и после "Прерывание по задержке времени".
2. Можно переименовать OB, установить язык программирования и выбрать номер блока. Перейти на ручное редактирование, если необходимо выбрать отличный от назначенного автоматически номер блока.
3. Отредактировать подпрограмму OB прерывания по задержке времени и вставить реакцию программы, которая должна последовать при наступлении события задержки по времени. Из OB прерывания по задержке времени могут вызываться другие блоки кода FC и FB. Максимальная глубина вложенности для программ обеспечения безопасности составляет четыре уровня. Для других программ максимальная глубина вложенности составляет шесть уровней.
4. Имена новых назначенных OB прерывания по задержке времени доступны при обработке параметра OB_NR инструкций SRT_DINT и CAN_DINT.

Параметр STATUS из QRY_DINT

Таблица 9-122 При наличии ошибки (REL_VAL <> 0) параметр STATUS = 0.

Бит	Значение	Описание
0	0	В режиме RUN
	1	При запуске
1	0	Прерывание активировано.
	1	Прерывание деактивировано.
2	0	Прерывание не активно или истекло.
	1	Прерывание активно.
4	0	Блок OB с указанным в OB_NR номером OB не существует.
	1	Блок OB с указанным в OB_NR номером OB существует.
Прочие биты		Всегда 0

Коды условий

Таблица 9-123 Коды условий для SRT_DINT, CAN_DINT и QRY_DINT

RET_VAL (W#16#...)	Описание
0000	Ошибки отсутствуют
8090	Неправильный параметр OB_NR

RET_VAL (W#16#...)	Описание
8091	Неправильный параметр DTIME
80A0	Прерывание по задержке времени еще не запущено.

9.5.5 Инструкции DIS_AIRT und EN_AIRT (задержка/активация выполнения прерываний с более высоким приоритетом и асинхронных ошибок)

Обработка прерываний может быть активирована и деактивирована с помощью инструкций DIS_AIRT и EN_AIRT.

Таблица 9-124 Инструкции DIS_AIRT и EN_AIRT

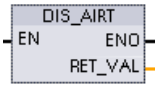
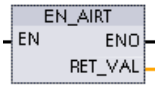
LAD/FBD	SCL	Описание
	`DIS_AIRT();`	DIS_AIRT откладывает обработку новых прерывающих событий. Возможно многократное выполнение DIS_AIRT в одном OB.
	`EN_AIRT();`	EN_AIRT активирует обработку прерывающих событий, которые прежде были деактивированы с помощью инструкции DIS_AIRT. Каждое выполнение DIS_AIRT должно быть удалено через выполнение EN_AIRT. Инструкции EN_AIRT должны быть выполнены в том же OB или в одном из вызванных из него FC / FB, прежде чем для этого OB снова будут активированы прерывания.

Таблица 9-125 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
RET_VAL	OUT	Int	Число задержке = число выполненных DIS_AIRT в листе ожидания.

Операционная система считает количество выполненных DIS_AIRT. Каждая из этих инструкций продолжает действовать до ее удаления с помощью инструкции EN_AIRT или до полного выполнения текущего OB. Пример: Если обработка прерывания была деактивирована пять раз через пять выполнений DIS_AIRT, то следует произвести отмену через пятикратное выполнение EN_AIRT, чтобы снова активировать обработку прерывания.

После повторной активации прерывающих событий обрабатываются прерывания, возникшие при выполнении DIS_AIRT, или прерывания будут обработаны сразу по выполнении текущего OB.

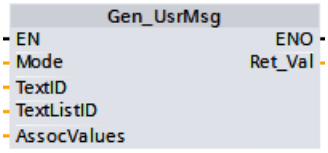
Параметр RET_VAL показывает, как часто деактивировалась обработка прерываний, т.е. сколько выполнений DIS_AIRT было внесено в лист ожидания. Обработка прерывания снова активируется, только если параметр RET_VAL = 0.

9.6 Прерывания

9.6.1 Gen_UsrMsg (создание диагностических сообщений пользователя)

Таблица 9-126 Инструкция Gen_UsrMsg

LAD/FBD	SCL	Описание

	`ret_val :=Gen_UsrMsg(`
	`  Mode:=_uint_in_,`
	`  TextID:=_uint_in_,`
	`  TextListID:=_uint_in_,`
	`  AssocValues:=_struct_inout_);`
		С помощью инструкции "Gen_UsrMsg" создается аварийное диагностическое сообщение пользователя, в случае которого речь может идти о наступающем или уходящем аварийном сообщении. С помощью аварийных диагностических сообщений пользователя можно внести запись пользователя в буфер диагностики и отправить соответствующее аварийное сообщение.
		Запись в буфер диагностики создается синхронно. Передача аварийного сообщения выполняется асинхронно.
		Если при выполнении инструкции возникает ошибка, то она выводится через параметр RET_VAL.

Содержание аварийного сообщения

Список текстов определяет содержание аварийного сообщения:

- Для определения используемого списка текстов используется параметр TextListID. С этой целью открыть в дереве проекта диалоговое окно "Списки текстов". Открыть в диалоговом окне "Списки текстов" столбец "ID". Применить ID в параметре TextListID.
- Выбрать через параметр TextID запись в списке текстов, которая должна быть внесена в буфер диагностики. Для этого выбрать в диалоговом окне "Записи в списках текстов" запись, используя для этого номер из столбца "Диапазон от/диапазон до" в параметре TextID. Для обоих столбцов "Диапазон от" и "Диапазон до" записей в списках текстов следует использовать один и тот же номер.

Дополнительная информация о списках текстов находится в информационной системе STEP 7.

Определение присваиваемых значений

Запись в списке текстов определяет новые присваиваемые значения, которые должны быть добавлены к аварийному сообщению:

- Добавить к записи в списке текстов следующую информацию для определения присваиваемых значений:
  @<номер присваиваемого значения><тип элемента><спецификация формата>@.
- Через тип системных данных AssocValues указывается, какое присваиваемое значение должно быть добавлено при создании аварийного сообщения.

Дополнительная информация о структуре присваиваемых значений находится в информационной системе STEP 7.

Параметр

В приведенной ниже таблице показаны параметры инструкции "Gen_UsrMsg":

Параметр	Объявление	Тип данных	Область памяти	Описание
Mode	Input	UInt	I, Q, M, D, L или константа	Параметры для выбора состояния аварийного сообщения: • 1: наступающее аварийное сообщение • 2: уходящее аварийное сообщение
TextID	Input	UInt	I, Q, M, D, L или константа	ID записи в списке текстов, которая должна использоваться для текста аварийного сообщения.
TextListID	Input	UInt	I, Q, M, D, L или константа	ID списка текстов, содержащего запись в списке текстов
Ret_Val	Return	Int	I, Q, M, D, L	Код ошибки инструкции.
AssocValues	InOut	VARIANT	D, L	Указатель на тип системных данных AssocValues, с помощью которого возможно определение присваиваемых значений.

Дополнительную информацию о допустимых типах данных можно найти в разделе "Типы данных (Страница 123)".

Параметр AssocValues

С помощью типа системных данных AssocValues определяется, какие присваиваемые значения будут передаваться. Возможно до восьми присваиваемых значений. Ввести тип данных "AssocValues" как блок данных, чтобы создать структуру.

Выбор присваиваемых значений выполняется путем ввода номеров присваиваемых значений для параметров Value[x]. См. примечание ниже:

- Инструкция Gen_UsrMsg обрабатывает значения для TextID и TextListID как отправляемые присваиваемые значения. Поэтому числа "1" и "2" назначены заранее для адресации присваиваемых значений. Не использовать числа "1" и "2" для адресации присваиваемых значений.
- Использовать для адресации присваиваемого значения в параметре Value [1] как номер "3", в параметр Value [2] как номер "4" и т. д.

Байт	Параметр	Тип данных	Начальное значение	Описание	Номер присваиваемого значения
0..1	Value[1]	UINT	0	Первое присваиваемое значение аварийного сообщения	3
2..3	Value[2]	UINT	0	Второе присваиваемое значение аварийного сообщения	4
4..5	Value[3]	UINT	0	...	5
6..7	Value[4]	UINT	0	...	6
8..9	Value[5]	UINT	0	...	7
10..11	Value[6]	UINT	0	...	8
12..13	Value[7]	UINT	0	...	9
14..15	Value[8]	UINT	0	Восьмое присваиваемое значение аварийного сообщения	10

Параметр RET_VAL

Таблица ниже определяет выходные значения для параметра RET_VAL . См. также Общие коды ошибок для расширенных инструкций (Страница 554).

Код ошибки* (W#16#...)	Значение
0000	Ошибки отсутствуют
8080	Значение в параметре MODE не поддерживается.
80C1	Недостаточное ресурсов из-за большого числа параллельных вызовов.
8528	Параметр 5 (AssocValues) не выровнен по байтам.
853A	Параметр 5 (AssocValues) ссылается на недопустимую точку.
*Коды ошибок могут отображаться в редакторе текстов программ в виде целых чисел или шестнадцатеричных значений.	

9.7 Диагностика (PROFINET или PROFIBUS)

9.7.1 Диагностические инструкции

Следующие диагностические инструкции могут использоваться с PROFINET или PROFIBUS:

- Инструкция RD_SINFO (Страница 440): Считывает начальную информацию текущего блока OB
- Инструкция LED (Страница 449): Считывает состояние светодиодных индикаторов распределенного периферийного устройства
- Инструкция Get_IM_Data (Страница 450): Проверяет данные идентификации и обслуживания (I&M) для указанного модуля или субмодуля
- Инструкция Get_Name (Страница 452): Считывает имя устройства PROFINET IO, ведомого устройства PROFIBUS или ведомого устройства AS-i
- Инструкция GetStationInfo (Страница 458): Считывает IP- или MAC-адрес устройства PROFINET IO в локальной системе ввода-вывода или устройства PROFINET IO в системе ввода-вывода нижнего уровня (подключенной через CP/CM модули).
- Инструкция DeviceStates (Страница 468): Опрашивает рабочие состояния распределенного периферийного устройства в подсистеме ввода-вывода
- Инструкция ModuleStates (Страница 474): Опрашивает рабочие состояния модулей у распределенного периферийного устройства
- Инструкция GET_DIAG (Страница 479): Считывает диагностическую информацию из указанного устройства

Примечание

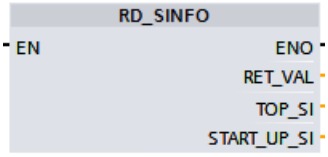
Инструкция GetStationInfo может использоваться только для устройств PROFINET IO. Нельзя использовать инструкцию с ведомыми устройствами PROFIBUS DP.

9.7.2 RD_SINFO (считывание начальной информации текущего блока OB)

Описание

Таблица 9-127 Инструкция RD_SINFO

LAD/FBD	SCL	Описание

	`ret_val := RD_SINFO(`
	`TOP_SI=>_variant_out_,`
	
	`START_UP_SI=>_variant_out_)`
	`;`
		С помощью инструкции "RD_SINFO" считывается начальная информация следующих OB:
		• Последнего вызванного блока OB, который еще не обработан полностью
		• Последнего пускового OB, который запустил CPU
		Отметка времени отсутствует в обоих случаях. Если вызов выполняется в OB 100, OB 101 или OB 102, возвращается два идентичных типа начальной информации.

Параметр

В приведенной ниже таблице показаны параметры инструкции "RD_SINFO":

Параметр	Объявление	Тип данных	Область памяти	Описание
RET_VAL	Return	INT	I, Q, M, D, L	Информация об ошибке
TOP_SI	Output	VARIANT	D, L	Начальная информация текущего блока OB
START_UP_SI	Output	VARIANT	D, L	Начальная информация последнего запущенного пускового OB

Дополнительную информацию о допустимых типах данных можно найти в главе "Типы данных (Страница 123)".

SDT параметра TOP_SI

В таблице ниже показаны возможные SDT для параметра TOP_SI:

Организационные блоки (OB)	Типы системных данных (SDT)	Номера типов системных данных
Любые	SI_classic*	592*
	SI_none	593
ProgramCycleOB	ProgramCycleOB	594
TimeOfDay	TimeOfDay	595
TimeDelayOB	SI_Delay	596
CyclicOB	SI_Cyclic	597
ProcessEventOB	SI_HWInterrupt	598

Организационные блоки (OB)	Типы системных данных (SDT)	Номера типов системных данных
ProfileEventOB StatusEventOB UpdateEventOB	SI_Submodule	601
SynchronousCycleOB	SI_SynchCycle	602
IOredundancyErrorOB	SI_IORedundancyError	604
CPUredundancyErrorOB	SI_CPURedundancyError	605
TimeErrorOB	SI_TimeError	606
DiagnosticErrorOB	SI_DiagnosticInterrupt	607
PullPlugEventOB	SI_PlugPullModule	608
PeripheralAccessErrorOB	SI_AccessError	609
RackStationFailureOB	SI_StationFailure	610
ServoOB	SI_Servo	611
IpoOB	SI_Ipo	612
StartupOB	SI_Startup	613
ProgrammingErrorOB IOaccessErrorOB	SI_ProgIOAccessError	614

*SDT SI_classic не относится к S7-1200. S7-1200 CPU возвращает RET_VAL из #16#8081, если параметр TOP_SI имеет тип SI_classic.

SDT параметра START_UP_SI

В таблице ниже показаны возможные SDT для параметра START_UP_SI:

Типы системных данных (SDT)	Номера типов системных данных
SI_classic*	592
SI_none	593
SI_Startup	613

*SDT SI_classic не относится к S7-1200. S7-1200 CPU возвращает RET_VAL из #16#8083, если параметр START_UP_SI имеет тип SI_classic.

Структуры

В следующей таблице представлено определение элементов отдельных структур:

Таблица 9-128 Структура SI_classic

Элемент структуры	Тип данных	Описание
EV_CLASS	BYTE	• Биты от 0 до 3: ID события • Биты от 4 до 7: Класс события
EV_NUM	BYTE	Номер события
PRIORITY	BYTE	Номер класса приоритетов (значение B#16#FE: OB недоступен или деактивирован или не может быть запущен в текущем рабочем состоянии)

Элемент структуры	Тип данных	Описание
NUM	BYTE	Номер OB
TYP2_3	BYTE	Идентификатор данных 2_3: Идентифицирует введенную в ZI2_3 информацию
TYP1	BYTE	Идентификатор данных 1: Идентифицирует введенную в ZI1 информацию
ZI1	WORD	Дополнительная информация 1
ZI2_3	DWORD	Дополнительная информация 2_3

Таблица 9-129 Структура SI_none

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"
OB_Nr	UINT	Номер OB (1 ... 32767)

Таблица 9-130 Структура SI_ProgramCycle

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT := 1	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"
OB_Nr	UINT	Номер OB (1 ... 32767)
Initial_Call	BOOL	Для OB_Class = 1, 30, 52, 61, 65
Remanence	BOOL	Для OB_Class = 1

Таблица 9-131 Структура SI_TimeOfDay

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT := 10	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"
OB_Nr	UINT	Номер OB (1 ... 32767)
CaughtUp	BOOL	Для OB_Class = 10
SecondTime	BOOL	Для OB_Class = 10

Таблица 9-132 Структура SI_Delay

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT := 20	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"
OB_Nr	UINT	Номер OB (1 ... 32767)
Sign	WORD	Для OB_Class = 20

Таблица 9-133 Структура SI_Cyclic

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT := 30	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"
OB_Nr	UINT	Номер OB (1 ... 32767)
Initial_Call	BOOL	Для OB_Class = 1, 30, 52, 61, 65
Event_Count	INT	Для OB_Class = 30, 51, 52, 61, 65, 91, 92

Таблица 9-134 Структура SI_HWInterrupt

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT := 40	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"
OB_Nr	UINT	Номер OB (1 ... 32767)
LADDR	HW_IO	Для OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
USI	WORD	Для OB_Class = 40
IChannel	USINT	Для OB_Class = 40
EventType	BYTE	Для OB_Class = 40

Таблица 9-135 Структура SI_Submodule

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"

Элемент структуры	Тип данных	Описание
OB_Nr	UINT	Номер OB (1 ... 32767)
LADDR	HW_IO	Для OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Slot	UINT	Для OB_Class = 55, 56, 57
Specifier	WORD	Для OB_Class = 55, 56, 57

Таблица 9-136 Структура SI_SynchCycle

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT := 61	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"
OB_Nr	UINT	Номер OB (1 ... 32767)
Initial_Call	BOOL	Для OB_Class = 1, 30, 52, 61, 65
PIP_Input	BOOL	Для OB_Class = 61, 91, 92
PIP_Output	BOOL	Для OB_Class = 61, 91, 92
IO_System	USINT	Для OB_Class = 61, 91, 92
Event_Count	INT	Для OB_Class = 30, 51, 52, 61, 65, 91, 92
SyncCycleTime	LTIME	Рассчитанное время цикла

Таблица 9-137 Структура SI_IORedundancyError

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT := 70	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"
OB_Nr	UINT	Номер OB (1 ... 32767)
LADDR	HW_ANY	Для OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Event_Class	BYTE	Для OB_Class = 70, 83, 85, 86
Fault_ID	BYTE	Для OB_Class = 70, 80, 83, 85, 86

Таблица 9-138 Структура SI_CPURedundancyError

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT := 72	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"

Элемент структуры	Тип данных	Описание
OB_Nr	UINT	Номер OB (1 ... 32767)
Switch_Over	BOOL	Для OB_Class = 72

Таблица 9-139 Структура SI_TimeError

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT := 80	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"
OB_Nr	UINT	Номер OB (1 ... 32767)
Fault_ID	BYTE	Для OB_Class = 70, 80, 83, 85, 86
Csg_OBnr	OB_ANY	Для OB_Class = 80
Csg_Prio	UINT	Для OB_Class = 80

Таблица 9-140 Структура SI_DiagnosticInterrupt

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT := 82	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"
OB_Nr	UINT	Номер OB (1 ... 32767)
IO_State	WORD	Для OB_Class = 82
LADDR	HW_ANY	Для OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Channel	UINT	Для OB_Class = 82
MultiError	BOOL	Для OB_Class = 82

Таблица 9-141 Структура SI_PlugPullModule

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT := 83	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"
OB_Nr	UINT	Номер OB (1 ... 32767)
LADDR	HW_IO	Для OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Event_Class	BYTE	Для OB_Class = 70, 83, 85, 86
Fault_ID	BYTE	Для OB_Class = 70, 80, 83, 85, 86

Таблица 9-142 Структура SI_AccessError

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT := 85	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"
OB_Nr	UINT	Номер OB (1 ... 32767)
LADDR	HW_IO	Для OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Event_Class	BYTE	Для OB_Class = 70, 83, 85, 86
Fault_ID	BYTE	Для OB_Class = 70, 80, 83, 85, 86
IO_Addr	UINT	Для OB_Class = 85
IO_LEN	UINT	Для OB_Class = 85

Таблица 9-143 Структура SI_StationFailure

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT := 86	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"
OB_Nr	UINT	Номер OB (1 ... 32767)
LADDR	HW_IO	Для OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Event_Class	BYTE	Для OB_Class = 70, 83, 85, 86
Fault_ID	BYTE	Для OB_Class = 70, 80, 83, 85, 86

Таблица 9-144 Структура SI_Servo

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT := 91	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"
OB_Nr	UINT	Номер OB (1 ... 32767)
Initial_Call	BOOL	Для OB_Class = 1, 30, 52, 61, 65
PIP_Input	BOOL	Для OB_Class = 61, 91, 92
PIP_Output	BOOL	Для OB_Class = 61, 91, 92
IO_System	USINT	Для OB_Class = 61, 91, 92
Event_Count	INT	Для OB_Class = 30, 51, 52, 61, 65, 91, 92
Synchronous	BOOL	

Таблица 9-145 Структура SI_Ipo

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT := 92	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"
OB_Nr	UINT	Номер OB (1 ... 32767)
Initial_Call	BOOL	Для OB_Class = 1, 30, 52, 61, 65
PIP_Input	BOOL	Для OB_Class = 61, 91, 92
PIP_Output	BOOL	Для OB_Class = 61, 91, 92
IO_System	USINT	Для OB_Class = 61, 91, 92
Event_Count	INT	Для OB_Class = 30, 51, 52, 61, 65, 91, 92
Reduction	UINT	Для OB_Class = 92

Таблица 9-146 Структура SI_Startup

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT := 100	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"
OB_Nr	UINT	Номер OB (1 ... 32767)
LostRetentive	BOOL	Для OB_Class = 100
LostRTC	BOOL	Для OB_Class = 100

Таблица 9-147 Структура SI_ProgIOAcessError

Элемент структуры	Тип данных	Описание
SI_Format	USINT	• 16#FF = информация отсутствует • 16#FE = оптимизированная начальная информация
OB_Class	USINT	Класс OB для "Информация отсутствует" или "Оптимизированная начальная информация"
OB_Nr	UINT	Номер OB (1 ... 32767)
BlockNr	UINT	Для OB_Class = 121, 122
Reaction	USINT	Для OB_Class = 121, 122
Fault_ID	BYTE	Для OB_Class = 121, 122
BlockType	USINT	Для OB_Class = 121, 122
Area	USINT	Для OB_Class = 121, 122
DBNr	DB_ANY	Для OB_Class = 121, 122
Csg_OBNr	OB_ANY	Для OB_Class = 121, 122
Csg_Prio	USINT	Для OB_Class = 121, 122
Width	USINT	Для OB_Class = 121, 122

Примечание

При создании со свойством блока "Стандартный" указанные для структуры SI_classic элементы по содержанию совпадают с временными переменными (тегами) OB.

Однако следует помнить, что временные переменные (теги) в отдельных блоках OB могут иметь другие имена и другие типы данных. Также следует помнить, что интерфейс вызова каждого OB содержит дополнительную информацию о дате и времени запроса OB.

Биты 4 - 7 элемента структуры EV_CLASS содержат класс события. Здесь возможны следующие значения:

- 1: Начальные события из стандартных блоков OB
- 2: Начальные события из синхронных OB обработки ошибок
- 3: Начальные события из асинхронных OB обработки ошибок

Элемент структуры PRIORITY возвращает класс приоритетов текущего блока OB.

Помимо этих двух элементов, значение имеет и NUM . NUM содержит номер текущего блока OB или последнего запущенного пускового OB.

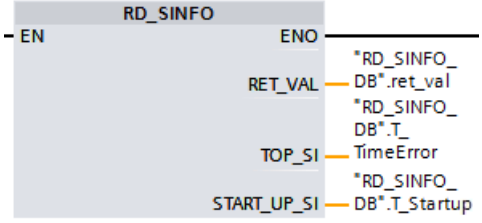
Параметр RET_VAL

В следующей таблице объясняются значения параметра RET_VAL:

Код ошибки* (W#16#...)	Значение
8081	Начальная информация текущего блока OB не соответствует указанному типу системных данных
8083	Начальная информация последнего запущенного пускового OB не соответствует указанному типу системных данных
*Коды ошибок могут отображаться в редакторе текстов программ в виде целочисленных значений или шестнадцатеричных значений.	

Пример

OB прерывания по ошибке времени (OB 80) это последний вызванный и еще не до конца обработанный OB. Пусковой OB (OB 100) это последний запущенный пусковой OB. Вызов инструкции для считывания пусковой информации имеет следующий вид, при этом RD_SINFO_DB - это блок данных, содержащий переменные (теги) SDT для типов OB:



В следующей таблице показано присвоение элементов структуры параметра TOP_SI инструкции RD_SINFO соответствующим локальными переменными (тегам) блока OB 80.

Элемент структуры TOP_SI	Тип данных	OB 80 - соответствующая локальная переменная (тег)	Тип данных
EV_CLASS	BYTE	OB80_EV_CLASS	BYTE
EV_NUM	BYTE	OB80_FLT_ID	BYTE
PRIORITY	BYTE	OB80_PRIORITY	BYTE
NUM	BYTE	OB80_OB_NUMBR	BYTE
TYP2_3	BYTE	OB80_RESERVED_1	BYTE
TYP1	BYTE	OB80_RESERVED_2	BYTE
ZI1	WORD	OB80_ERROR_INFO	WORD
ZI2_3	DWORD	OB80_ERR_EV_CLASS	BYTE
		OB80_ERR_EV_NUM	BYTE
		OB80_OB_PRIORITY	BYTE
		OB80_OB_NUM	BYTE

В следующей таблице показано присвоение элементов структуры параметра START_UP_SI инструкции RD_SINFO соответствующим локальными переменными (тегам) блока OB 100.

Элемент структуры START_UP_SI	Тип данных	OB 100 - локальная переменная (тег)	Тип данных
EV_CLASS	BYTE	OB100_EV_CLASS	BYTE
EV_NUM	BYTE	OB100_STRTUP	BYTE
PRIORITY	BYTE	OB100_PRIORITY	BYTE
NUM	BYTE	OB100_OB_NUMBR	BYTE
TYP2_3	BYTE	OB100_RESERVED_1	BYTE
TYP1	BYTE	OB100_RESERVED_2	BYTE
ZI1	WORD	OB100_STOP	WORD
ZI2_3	DWORD	OB100_STRT_INFO	DWORD

9.7.3 LED (чтение состояния светодиода)

Таблица 9-148 Инструкция LED

LAD/FBD	SCL	Описание

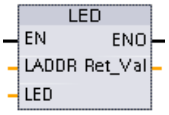
	`ret_val := LED(`
	`  laddr:=_word_in_,`
	`  LED:=_uint_in_);`
		С помощью инструкции LED считывается состояние светодиодных индикаторов на CPU (Страница 1387). Указанное состояние светодиода возвращается выходом RET_VAL.

Таблица 9-149 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание		
LADDR	IN	HW_IO	Идентификатор CPU[1]		
LED	IN	UInt	Идентификационный номер светодиода		
			1	RUN/STOP	Цвет 1 = зеленый, цвет 2 = желтый
			2	Ошибка	Цвет 1 = красный
			3	Обслуживание	Цвет 1 = желтый
RET_VAL	OUT	Int	Состояние светодиода		

[1] Для идентификации подключенного модуля CPU выбрать Local~Common из выпадающего списка параметра.

Таблица 9-150 Состояние RET_VAL

RET_VAL (W#16#...)	Описание	
Состояние светодиода от 0 до 9	0	Светодиод отсутствует
	1	Выкл
	2	Цвет 1 вкл (включен постоянно)
	3	Цвет 2 вкл (включен постоянно)
	4	Цвет 1 мигает с частотой 2 Гц
	5	Цвет 2 мигает с частотой 2 Гц
	6	Цвета 1 и 2 мигают попеременно с частотой 2 Гц
	9	Состояние светодиода недоступно
8091	Указанное LADDR устройство отсутствует	
8092	Указанное LADDR устройство не поддерживает светодиоды	
8093	Светодиодная идентификация не определена	
80Bx	Указанный LADDR модуль CPU не поддерживает инструкцию LED	

9.7.4 Get_IM_Data (чтение данных идентификации и обслуживания)

Инструкция Get_IM_Data проверяет данные идентификации и обслуживания (I&M) для указанного модуля или субмодуля.

Таблица 9-151 Инструкция Get_IM_Data

LAD/FBD	SCL	Описание

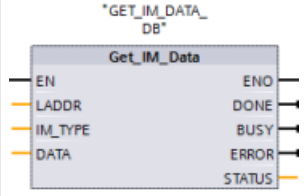
	`"GET_IM_DATA_DB"(LADDR:=16#0 ,`
	`IM_TYPE:=0,`
	`DONE=>_bool_out_,`
	`BUSY=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`DATA:=_variant_inout_);`
		Инструкция Get_IM_Data проверяет данные идентификации и обслуживания (I&M) для указанного модуля или субмодуля.

Таблица 9-152 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
LADDR	Input	HW_EA	Идентификатор модуля
IM_TYPE	Input	UInt	Номер данных идентификации и обслуживания (I&M): • 0: I&M0 (MLFB, серийный номер, версия и другая информация) • 1: I&M1 (десигнаторы) • 2: I&M2 (дата установки) • 3: I&M3 (дескриптор) • 4: I&M4 (сигнатура)
RET_VAL	Output	Int	Статус (код условия)
DATA	InOut	Variant	Данные I&M (STRING или массив из BYTE); рекомендуется использовать SDT "IM0_Data" дляIM_TYPE = 0.

С помощью данных идентификации и обслуживания (I&M) можно проверить конфигурацию системы, обнаружить изменения в оборудовании просмотреть данные о техническом обслуживании. Идентификационные данные модуля (I-данные) – только для чтения. Эксплуатационные данные модуля (M-данные) зависят от системной информации, например даты установки. M-данные создаются при планировании технического обслуживания и записываются в модуль:

- Если в случае типа данных параметра DATA речь идет о символьной строке, то текущая длина строки задается в соответствии с длиной I&M данных.
- Если в случае типа данных параметра DATA речь идет о массиве из Byte или Char, то I&M данные копируются как последовательность байтов.
- Если в случае типа данных параметра DATA речь идет о структуре, то I&M данные копируются как последовательность байтов.
- Если соответствующий массив из Byte/Char вDATA длиннее, чем запрошенные I&M данные, то прикрепляется значение байта 16#00.
- Другие типы данных не поддерживаются, возвращается ошибка 8093.

Таблица 9-153 Коды условий

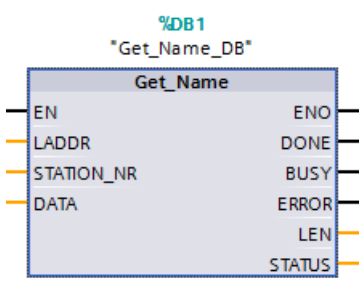
RET_VAL (W#16#...)	Описание
0	Ошибки отсутствуют
8091	LADDR отсутствует
8092	LADDR не адресует аппаратного объекта, поддерживающего I&M данные
8093	Указанный в параметре DATA тип данных не поддерживается
80B1	Инструкция DATA не поддерживается CPU для этого параметра LADDR
80B2	IM_TYPE не поддерживается CPU
8452	Вся I&M информация не вмещается в указанную в параметре DATA переменную. Частичный результат до байтовой длины переменной возвращается.

9.7.5 Get_Name (чтение имени устройства PROFINET IO)

Инструкция "Get_Name" считывает имя устройства PROFINET IO, ведомого устройства PROFIBUS или ведомого устройства AS-i. Имя модуля отображается на вкладке сети и в свойствах устройства ввода-вывода.

Таблица 9-154 Инструкция Get_Name

LAD/FBD	SCL	Описание

	`"Get_Name_DB"(`
	`  LADDR:=_uint_in_,`
	`  STATION_NR:=_uint_in_,`
	`  DONE=>_bool_out_,`
	`  BUSY=>_bool_out_,`
	`  ERROR=>_bool_out_,`
	`  LEN=>_dint_out_,`
	`  STATUS=>_word_out_,`
	`  DATA:=_variant_inout_);`
		С помощью инструкции "Get_Name " выполняется чтение имени устройства PROFINET IO или ведомого устройства PROFIBUS.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "Get_Name_DB" - это имя DB экземпляра.

Выбрать IO-устройство через аппаратный идентификатор системы распределенного ввода-вывода (в параметре LADDR), и номер устройства PROFINET IO или адрес PROFIBUS ведомого устройства PROFIBUS (в параметре STATION_NR).

После выполнения инструкции программа записывает имя IO устройства в адресованную параметром DATA область.

Считываемое имя зависит от типа IO-устройства:

- DP-Slave или IO-устройство: Имя головного модуля
- I-Slave или I-устройство: Имя интерфейсного модуля
- HMI панель: Имя интерфейса
- Компьютерная станция: Имя интерфейсного модуля
- Устройства GSD: Отображается имя точки доступа к устройству (Device Access Point, DAP) (имя интерфейса или головного модуля)

Инструкция записывает длину имени в параметр LEN. Если имя длиннее, чем указанная в параметре DATA область, то программа записывает только сегмент, соответствующий максимальной длине адресованной области.

Максимальная длина для имени составляет 128 символов.

Примечание

Считывание имени CPU (версия 1.1)

Если обоим параметрам LADDR и STATION_NR, присваивается значение 0, то инструкция записывает имя CPU.

Параметр

В приведенной ниже таблице показаны параметры инструкции Get_Name:

Параметр	Объявление	Тип данных	Описание
LADDR	IN	HW_IOSYSTEM	Аппаратный идентификатор (HW-IoSystem) системы распределенного ввода-вывода. Номер запрашивается из системных констант или свойств системы ввода/вывода.
STATION_NR	IN	UInt	• Устройство PROFINET IO: Номер устройства передается из свойств IO-устройства в "Адреса Ethernet" в Просмотр сетевых соединений. • Ведомое устройство PROFIBUS: Адрес PROFIBUS передается из свойств ведомого устройства PROFIBUS в "Адрес PROFIBUS" в Просмотр сетевых соединений.
DATA	IN_OUT	Variant	Указатель на область, в которую записывается имя.
DONE	OUT	Bool	Инструкция успешно выполняется. Имя модуля передается в область в параметре DATA.
BUSY	OUT	Bool	Параметр состояния: • 0: Выполнение инструкции завершено. • 1: Выполнение инструкции еще не завершено.
ERROR	OUT	Bool	Параметр состояния: • 0: Ошибки отсутствуют • 1: Во время выполнения инструкции произошла ошибка. Параметр STATUS содержит подробную информацию.
LEN	OUT	DInt	Длина имени IO-устройства (количество символов).
STATUS	OUT	Word	Параметр состояния: Параметр устанавливается только на время одного вызова. Поэтому для отображения состояния параметр STATUS необходимо скопировать в свободную область данных.

Параметр STATUS

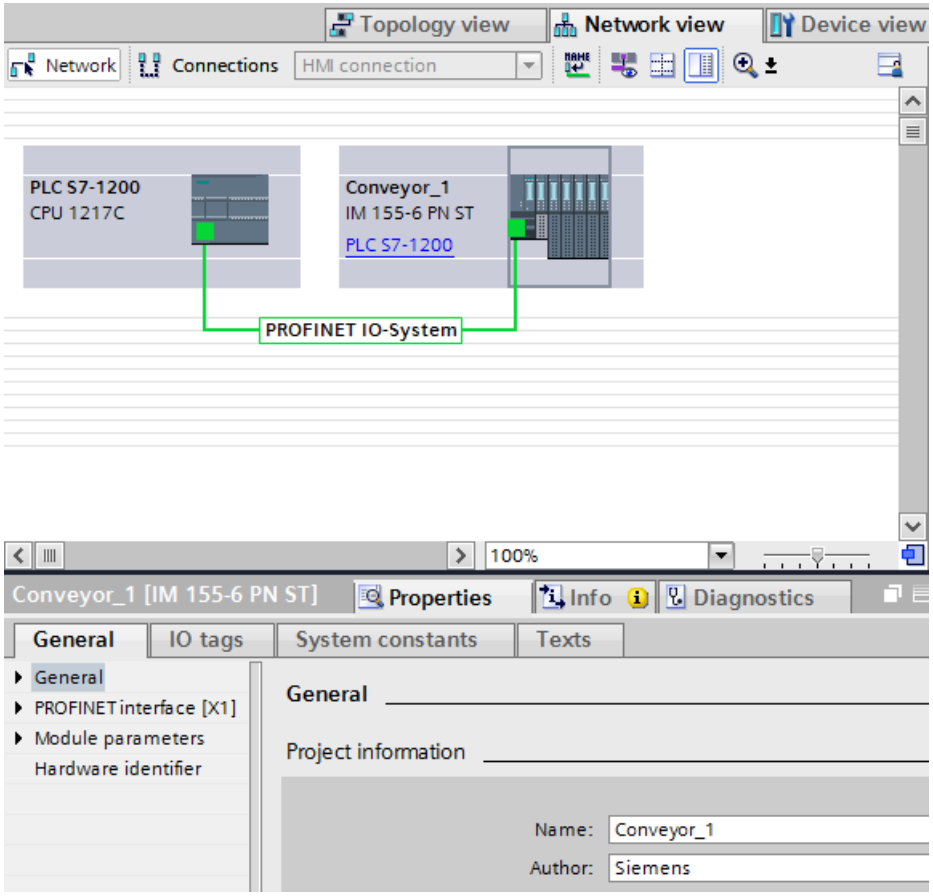
Код ошибки* (W#16#...)	Значение
0	Ошибки отсутствуют
7000	Обрабатываемые задания отсутствуют.
7001	Первый вызов асинхронной инструкции Get_Name Выполнение инструкции еще не завершено (BUSY = 1, DONE = 0).
7002	Дополнительный вызов асинхронной инструкции Get_Name. Выполнение инструкции еще не завершено (BUSY = 1, DONE = 0).
8090	Указанный в параметре LADDR аппаратный идентификатор отсутствует в проекте.
8092	Значение в параметре LADDR не адресует систему PROFINET-IO.
8093	Инструкция не поддерживает тип данных в параметре DATA.
8095	Номер устройства (параметр STATION_NR) отсутствует в выбранной системе PROFINET IO или не адресует IO-устройство.
80B1	Используемый CPU не поддерживает инструкцию.

Код ошибки* (W#16#...)	**Значение**
80C3	Временная ошибка ресурсов: В настоящий момент CPU обрабатывается максимально возможное число одновременных вызовов блоков. Инструкция Get_Name сможет быть выполнена только после завершения обработки минимум одного из вызовов блоков.
8852	Область, указанная в параметре DATA, слишком короткая для полного имени IO-устройства. Имя может быть записано до макс. возможной длины. Для считывания полного имени, следует использовать более длинную область данных в параметре DATA. Число символов в области как минимум должно быть равно числу символов, заданному параметром LEN.
* Коды ошибок могут отображаться в редакторе текстов программ в виде целых или шестнадцатеричных значений.	

Пример

Следующий пример показывает, как можно прочесть имя ET 200SP устройства PROFINET IO:

1. Конфигурирование ET200SP:
   - Создать ET 200SP с именем станции "Conveyor_1" на вкладке просмотра сетевых соединений и назначить ее той же системе PROFINET IO, что и CPU.
   - Назначить CPU на роль IO-контроллера для ET 200SP zu.
   - Следует использовать номер устройства по умолчанию "1" в свойствах в пункте "Адреса Ethernet".



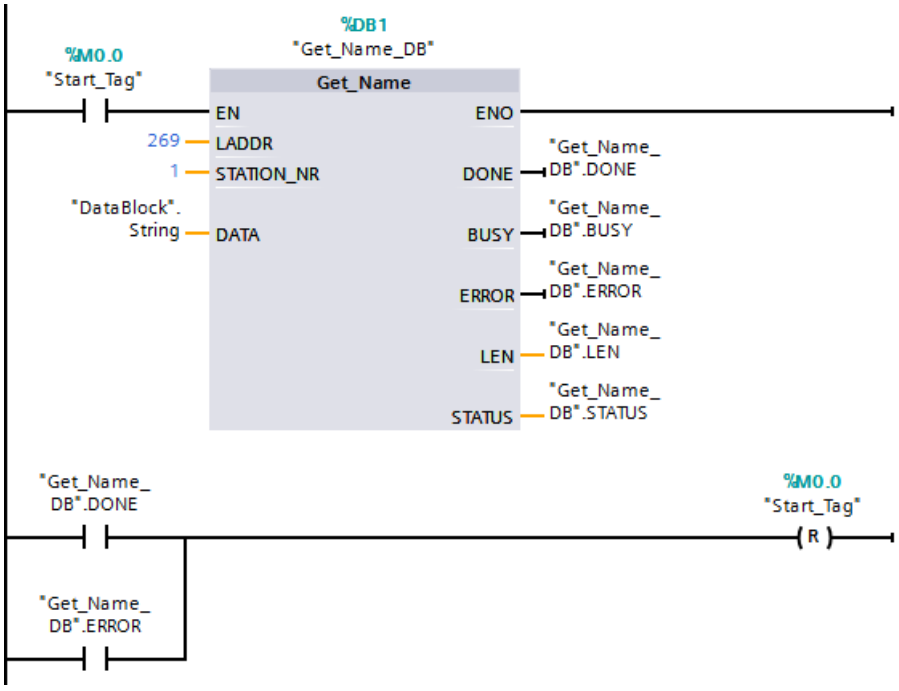
2. Назначение параметров для инструкции Get_Name:
   - Ввести аппаратный идентификатор системы ввода-вывода в параметре LADDR. В этом примере аппаратный идентификатор "269". Аппаратный идентификатор находится в следующем месте:
     Переменные (теги) PLC > Показать все переменные (теги) > Вкладка "Системные константы" > Локальная система PROFINET IO.
   - Ввести номер устройства ET200SP в параметре STATION_NR. Номер устройства в этом примере "1".

– Соединить переменную (тег) с типом данных STRING блока данных в параметре DATA.

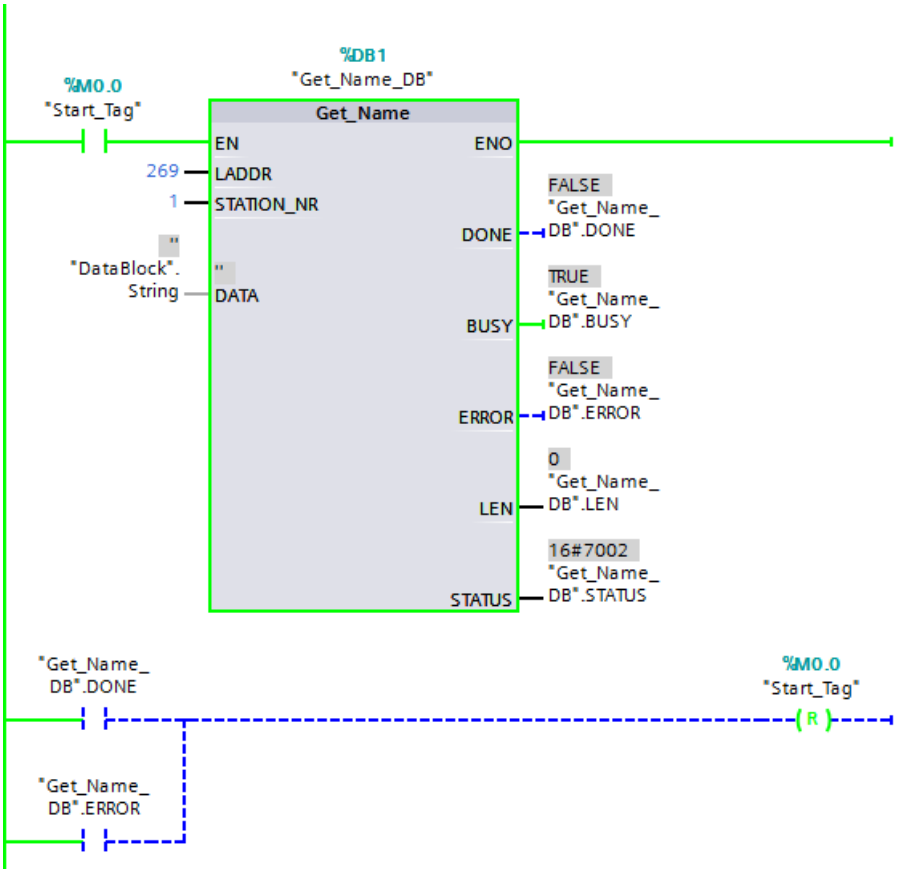
Примечание

Если при конфигурировании переменных (тегов) в параметре DATA для выбора используется выпадающий список, то следует выбрать DB (в примере "Блок данных") и переменную (тег) (в примере "String[ ]"). Для чтения всего типа данных String необходимо удалить скобки для получения следующего результата: "Блок данных".String.

– Определить переменные (теги) PLC (область памяти, меркеры) для выходных параметров инструкции.

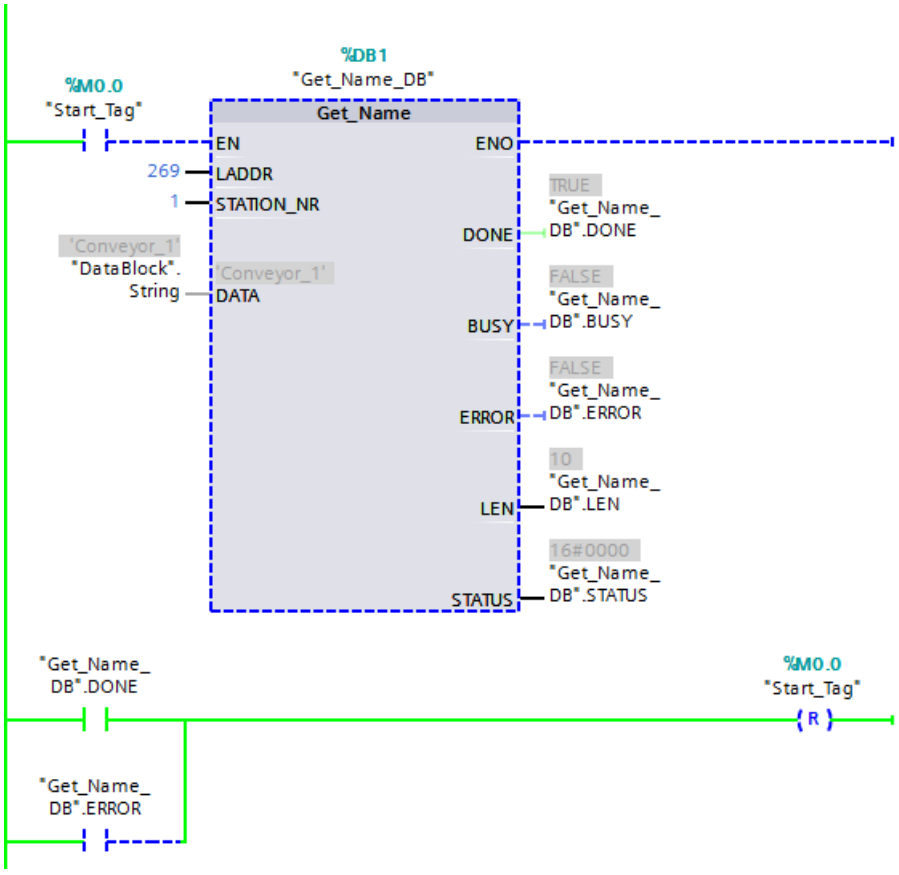


3. Выполнить инструкцию Get_Name:
   - При выполнении инструкции выходной параметр BUSY может быть установлен на 1, после чего параметр DONE устанавливается на 0.
   - Информация о коде ошибки отображается на выходном параметре STATUS.



4. Завершить выполнение инструкции Get_Name:
   - После выполнения инструкции программа "Conveyor_1" записывает имя станции ET200SP в блок данных в параметре DATA.

– Программа записывает "10", количество символов в имени станции, в параметр LEN.



9.7.6 GetStationInfo (чтение IP- или MAC-адреса устройства PROFINET IO)

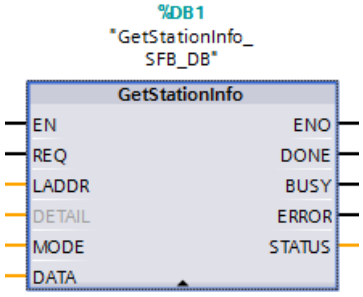
Инструкция "GetStationInfo" считывает IP- или MAC-адрес устройства PROFINET IO в локальной системе ввода-вывода или устройства PROFINET IO в системе ввода-вывода нижнего уровня (подключенной через CP/CM модули).

Примечание

Инструкция GetStationInfo может использоваться только для устройств PROFINET IO. Нельзя использовать инструкцию с ведомыми устройствами PROFIBUS DP.

Таблица 9-155 Инструкция GetStationInfo

LAD/FBD	SCL	Описание

	`"GetStationInfo_SFB_DB"(`
	`REQ:=_bool_in_,`
	`LADDR:=_uint_in_,`
	`DETAIL:=_uint_in_,`
	`MODE:=_uint_in_,`
	`DONE=>_bool_out_,`
	`BUSY=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`DATA:=_variant_inout_);`
		С помощью инструкции GetStationInfo считываются IP- или MAC-адрес устройства PROFINET IO. С помощью инструкции также возможно считывание IP- или MAC-адреса IO-устройства в системе ввода-вывода нижнего уровня (подключенной через CP/CM модули).

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "GetStationInfo_SFB_DB" - это имя DB экземпляра.

Для адресации IO-устройства используется аппаратный идентификатор станции в параметре LADDR. Аппаратный идентификатор находится в следующем месте: Переменные (теги) PLC > Показать все переменные (теги) > Вкладка "Системные константы". Искать IO-устройство в столбце "Имя" или "Hw_Device" в столбце "Тип данных".

С помощью параметра MODE выбирается считываемая информация.

В параметре DATA назначается область данных, в которую инструкция записывает считанные адресные данные. Для сохранения IP-адреса следует использовать структуру "IF_CONF_v4". Для сохранения MAC-адреса следует использовать структуру "IF_CONF_MAC".

Чтение адресных данных активируется через управляющий параметр REQ. Для этого требуется доступ к IO-устройству.

Инструкция показывает состояние выполнения для задания чтения через выходные параметрыBUSY, DONE и ERROR и выходной параметр STATUS.

Примечание

Для адресации IO-устройства следует использовать только аппаратный идентификатор станции.

У станции, IO-устройства и PROFINET-интерфейса есть свой аппаратный идентификатор у каждого. Для инструкции GetStationInfo следует использовать только аппаратный идентификатор станции.

При адресации, например, PROFINET интерфейса через параметр LADDR, адресные данные не считываются и CPU создает код ошибки "8092".

Для считывания адресных данных встроенного интерфейса PROFINET или CM/CP модуля в центральной конфигурации, следует использовать инструкцию "RDREC".

Параметр

В приведенной ниже таблице показаны параметры инструкции GetStationInfo:

Параметр	Объявление	Тип данных	Описание
REQ	IN	Bool	Управляющий параметр REQUEST Активирует считывание информации с REQ = 1.
LADDR	IN	HW_DEVICE	Аппаратный идентификатор станции IO-устройства Номер можно узнать в свойствах станции на вкладке сети или на вкладке "Системные константы" в стандартной таблице переменных (тегов).
DETAIL	IN	HW_SUBMODULE	Этот параметр DETAIL не используется. Не подключать параметр.
MODE	IN	UNIT	Выбор считываемых адресных данных: • MODE = 1: Адресный параметр по IPv4 (S7-1200 CPU с версией прошивки 4.2) • MODE = 2: MAC-адрес (S7-1200 CPU с версией прошивки 4.2)
DATA	IN_OUT	Variant	Указатель на область, в которую программа записывается адресные данные IO-устройства. Использовать структуру "IF_CONF_v4" для MODE = 1 и структуру "IF_CONF_MAC" для MODE = 2.
DONE	OUT	Bool	Программа успешно выполнила инструкцию. Программа передала адресные данные в параметр DATA.
BUSY	OUT	Bool	Параметр STATUS: • 0: Выполнение инструкции завершено. • 1: Выполнение инструкции еще не завершено.
ERROR	OUT	Bool	Параметр STATUS: • 0: Ошибки отсутствуют. • 1: Во время выполнения инструкции произошла ошибка. Подробная информация выводится в параметре STATUS.
STATUS	OUT	Word	Параметр STATUS: Параметр устанавливается только на время одного вызова. Поэтому для отображения состояния параметр STATUS необходимо скопировать в свободную область данных.

Параметр DATA

- Следует использовать структуру "IF_CONF_v4" в параметре DATA для сохранения адресного параметра в IPv4:

Байт	Параметр	Тип данных	Начальное значение	Описание
0 ... 1	Id	UINT	30	ID структуры "IF_CONF_v4"
2 ... 3	Length	UNIT	18	Длина считанных данных в BYTE
4 ... 5	Mode	UNIT	0	Не имеет значения для инструкции "GetStationInfo" (остается 0)

Байт	Параметр	Тип данных	Начальное значение	Описание
6 ... 9	InterfaceAddress	ARRAY [1..4] of BYTE	-	IP-адрес IO-устройства в формате IP_V4 (пример: 192.168.3.10): • addr[1] = 192 • addr[2] = 168 • addr[3] = 3 • addr[4] = 10
10 ... 13	SubnetMask	ARRAY [1..4] of BYTE	-	Маска подсети IO-устройства в формате IP_V4 (пример: 255.255.255.0): • addr[1] = 255 • addr[2] = 255 • addr[3] = 255 • addr[4] = 0
14 ... 17	DefaultRouter	ARRAY [1..4] of BYTE	-	IP-адрес маршрутизатора в формате IP_V4 (пример: 192.168.3.1): • addr[1] = 192 • addr[2] = 168 • addr[3] = 3 • addr[4] = 1

- Для сохранения MAC-адреса следует использовать структуру "IF_CONF_MAC" в параметре DATA.

Байт	Параметр	Тип данных	Начальное значение	Описание
0 ... 1	Id	UINT	3	ID структуры "IF_CONF_MAC"
2 ... 3	Length	UNIT	12	Длина считанных данных в BYTE
4 ... 5	Mode	UNIT	0	Не имеет значения для инструкции "GetStationInfo" (остается 0)
6 ... 11	MACAddress	ARRAY [1..6] of BYTE	-	MAC-адрес IO-устройства (пример: 08-00-06-12-34-56): • Mac[1] = 8 • Mac[2] = 0 • Mac[3] = 6 • Mac[4] = 12 • Mac[5] = 34 • Mac[6] = 56

Параметр STATUS

Код ошибки* (W#16#...)	Значение
0	Ошибки отсутствуют
7000	Обрабатываемые задания отсутствуют.

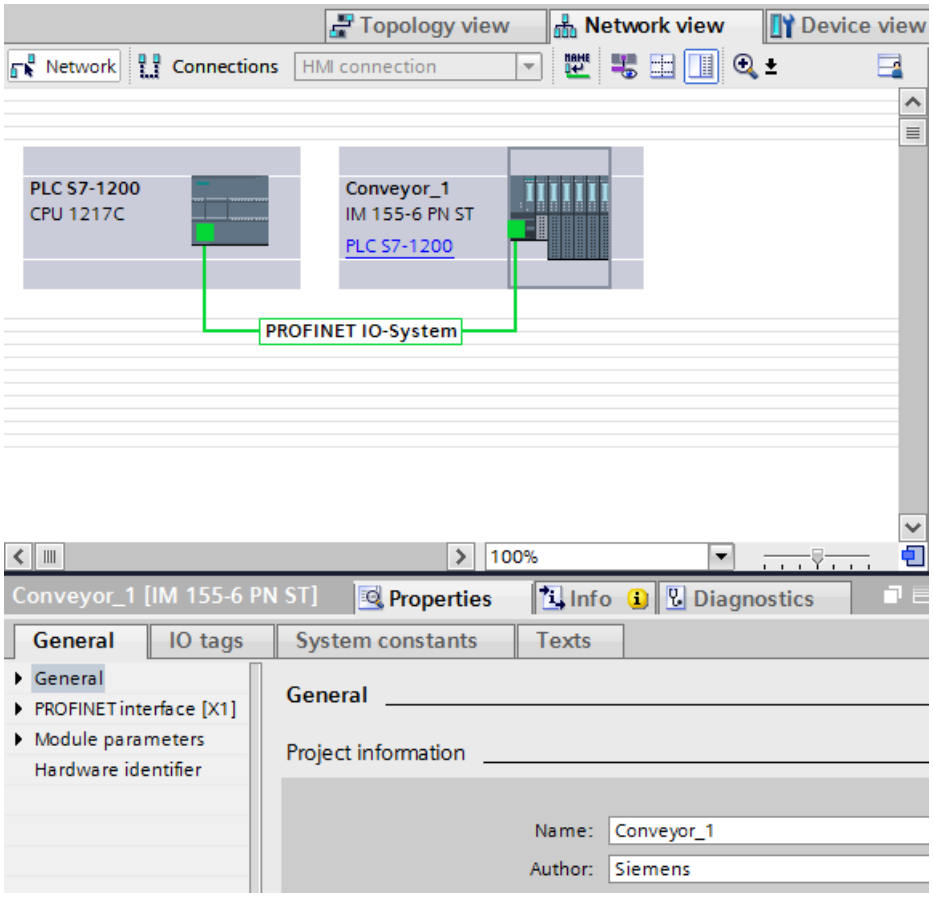
Код ошибки* (W#16#...)	**Значение**
7001	Первый вызов асинхронной инструкции GetStationInfo. Выполнение инструкции еще не завершено (BUSY = 1, DONE = 0).
7002	Дополнительный вызов асинхронной инструкции GetStationInfo. Выполнение инструкции еще не завершено (BUSY = 1, DONE = 0).
8080	Значение в параметре MODE не поддерживается.
8090	Указанный в параметре LADDR аппаратный идентификатор не сконфигурирован.
8092	Параметр LADDR не адресует устройство PROFINET IO.
8093	Недействительный тип данных в параметре DATA.
80A0	Запрошенная информация не считывается.
80C0	Адресованное IO-устройство недоступно.
80C3	Максимальное число одновременных вызовов инструкции GetStationInfo (10 экземпляров) достигнуто.
* Коды ошибок отображаются в редакторе текстов программ в виде целочисленных значений или как шестнадцатеричные значения.	

Пример

В следующем примере с помощью инструкции GetStationInfo считываются данные IP-адреса IO-устройства и выполняется запись информации в блок данных. Данные IP-адреса включают в себя IP-адрес, маску подсети и (если используется) адресные данные маршрутизатора.

IO-контроллер выполняет инструкцию GetStationInfo, а инструкция считывает данные IP-адреса IO-устройства нижнего уровня (в данном примере это ET200SP):

1. Конфигурирование ET200SP:
   - Создать ET 200SP с именем станции "Conveyor_1" на вкладке просмотра сетевых соединений и назначить ее той же системе PROFINET IO, что и CPU.
   - Назначить CPU на роль IO-контроллера для ET 200SP zu.



2. Назначение параметров для инструкции GetStationInfo :

– Создать пять переменные (тегов) и структуру с типом данных IF_CONF_v4 в глобальном блоке данных для хранение данных IP-адреса. Присвоить структуре любое имя. (имя структуры в примере "IP_Address")

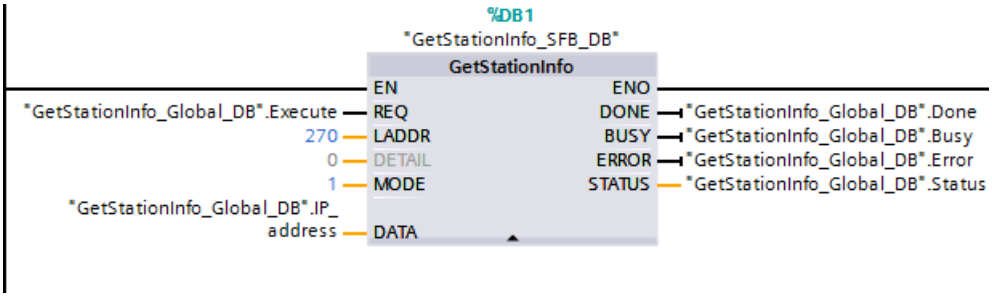
GetStationInfo_Global_DB

	Name	Data type	Start value
1	Static		
2	Execute	Bool	false
3	IP_address	IF_CONF_v4	
4	Id	UInt	30
5	Length	UInt	18
6	Mode	UInt	0
7	InterfaceAddress	IP_V4	
8	ADDR	Array[1..4] of Byte	
9	ADDR[1]	Byte	16#0
10	ADDR[2]	Byte	16#0
11	ADDR[3]	Byte	16#0
12	ADDR[4]	Byte	16#0
13	SubnetMask	IP_V4	
14	DefaultRouter	IP_V4	
15	Done	Bool	false
16	Busy	Bool	false
17	Error	Bool	false
18	Status	Word	16#0

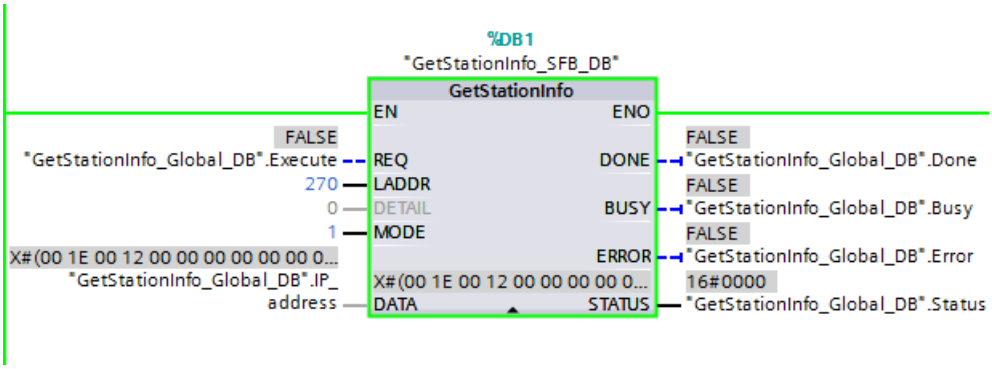
3. Назначение параметров для инструкции GetStationInfo:
   - Ввести аппаратный идентификатор IO-устройства в параметре LADDR. Аппаратный идентификатор обеспечивает однозначную идентификацию проекта. В этом примере аппаратный идентификатор "270". Аппаратный идентификатор находится в следующем месте: Переменные (теги) PLC > Показать все переменные (теги) > Вкладка "Системные константы".
     Искать IO-устройство в столбце "Имя" и "Hw_Device" в столбце "Тип данных". Назначенное значение это аппаратный идентификатор, который вводится в параметре LADDR.
   - Выбрать для параметра MODE значение 1 (чтение адресных параметров по IPv4).
   - Подключить структуру IF_CONF_v4 в параметре DATA.

     Примечание

     Если при конфигурировании переменных (тегов) в параметре DATA для выбора используется выпадающий список, то следует выбрать DB (в примере "GetStationInfo_Global_DB") и переменную (тег) (в примере "IP address"). Для чтения всего типа данных IF_CONF_v4, следует удалить пункт, следующий за "IP address" для получения следующего результата: "GetStationInfo_Global_DB".IP address
   - Определить переменные (теги) PLC своего глобально блока данных для выходных параметров инструкции.

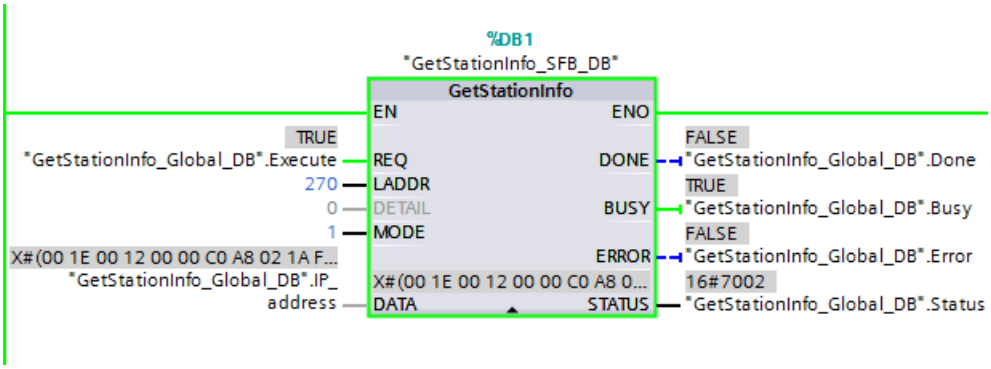


4. Выполнить инструкцию GetStationInfo:
   – Если вход REQ = 1 (FALSE), то инструкция не отображает на входном/выходном параметре DATA данных IP-адреса и на выходном параметре STATUS информацию о коде ошибки.



5. Завершить выполнение инструкции GetStationInfo:

– Если вход REQ = 1 (TRUE), то программа выполняет инструкцию и записывает IP-адрес в блок данных. Программа записывает IP-адрес, "C0 A8 02 1A" (десятичное значение "192.168.2.26"), во входной/выходной параметр DATA.



GetStationInfo_Global_DB

	Name	Data type	Start value	Monitor value
1	Static			
2	Execute	Bool	false	TRUE
3	IP_address	IF_CONF_v4		
4	Id	UInt	30	30
5	Length	UInt	18	18
6	Mode	UInt	0	0
7	InterfaceAddress	IP_V4		
8	ADDR	Array[1..4] of Byte		
9	ADDR[1]	Byte	16#0	16#C0
10	ADDR[2]	Byte	16#0	16#A8
11	ADDR[3]	Byte	16#0	16#02
12	ADDR[4]	Byte	16#0	16#1A
13	SubnetMask	IP_V4		
14	DefaultRouter	IP_V4		
15	Done	Bool	false	TRUE
16	Busy	Bool	false	FALSE
17	Error	Bool	false	FALSE
18	Status	Word	16#0	16#0000

9.7.7 Инструкция DeviceStates

Инструкция DeviceStates используется для вывода состояния всех ведомых устройств распределенного ввода/вывода, подключенных к специальному ведущему устройству распределенного ввода/вывода.

Таблица 9-156 Инструкция DeviceStates

LAD/FBD	SCL	Описание

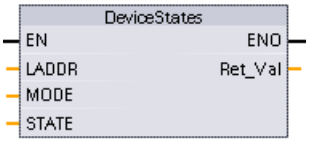
	`ret_val := DeviceStates(`
	`  laddr:=_word_in_,`
	`  mode:=_uint_in_,`
	`  state:=_variant_inout_);`
		DeviceStates опрашивает рабочие состояния устройств ввода-вывода в подсистеме ввода-вывода. После выполнения параметр STATE содержит состояние ошибки каждого отдельного устройства ввода/вывода в списке битов (для назначенных параметров LADDR и MODE). Эта информация соответствует состоянию устройства в окне диагностики STEP 7.
		Вход LADDR инструкции DeviceStates использует аппаратный идентификатор интерфейса распределенного ввода-вывода. В TIA Portal аппаратный идентификатор для PLC можно найти через типы данных "HW_IOSYSTEM" на вкладке системных констант в таблиц переменных (тегов) PLC.

Таблица 9-157 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
LADDR	IN	HW_IOSYSTEM	Логический адрес: (идентификатор системы ввода/вывода)
MODE	IN	UInt	Поддерживает пять режимов работы. Вход MODE устанавливает, какие данные будут выводиться в указанном для информации STATE месте. Возможны следующие режимы работы: • 1: Конфигурация устройства активна • 2: Устройство неисправно • 3: Устройство деактивировано • 4: Устройство существует • 5: Проблема в устройстве

Параметр и тип данных		Тип данных	Описание
RET_VAL	OUT	Int	Условие выполнения
STATE[1]	InOut	Variant	Буфер, принимающий состояние ошибки от каждого устройства: В качестве типа данных, для параметра STATE может быть выбран любой тип битовых данных (Bool, Byte, Word или DWord) или массив битовых данных. • Бит 0 первого байта выводимых данных STATE - это сводный бит. Если он установлен на TRUE, то это означает, что доступны и другие данные. • Выводимые параметром STATE данные показывают полное соответствие между позицией бита и адресом распределенного ввода-вывода. Эта адресация устройства TRUE для PROFIBUS и PROFINET. Пример: Бит 4 в первом байте совпадает с PROFIBUS-адресом 4 или номеромPROFINET устройства 4.

[1] У PROFIBUS DP длина информации о состоянии составляет 128 бит. У PROFINE I/O длина составляет 1024 бита.

После выполнения параметр STATE содержит состояние ошибки каждого отдельного устройства ввода/вывода как список битов (для назначенных параметров LADDR и MODE).

Таблица 9-158 Коды условий

RET_VAL (W#16#...)	Описание
0	Ошибки отсутствуют
8091	LADDR отсутствует.
8092	LADDR не обращается к системе ввода-вывода.
8093	Недействительный тип данных для параметра STATE: Действительные типы данных (Bool, Byte, Word или Dword) или массив из (Bool, Byte, Word или Dword)
80Bx	Инструкция DeviceStates не поддерживается CPU для этого параметра LADDR.
8452	Полные данные о состоянии являются слишком большими для назначенного параметра STATE . Буфер STATE содержит частичный результат.

9.7.7.1 Примеры для конфигурации DeviceStates

Пример в PROFIBUS

Пример в PROFIBUS состоит из следующих компонентов:

- 16 PROFIBUS-устройств с именами "DPSlave_10" до "DPSlave_25"
- 16 PROFIBUS-устройств используют по одному из PROFIBUS-адресов от 10 до 25 каждый.
- Для каждого ведомого устройства сконфигурировано несколько модулей ввода-вывода.
- Отображаются первые четыре байта выводимой информации параметра STATE.

MODE	Пример 1: Штатный режим работы без ошибок	Пример 2:Ведомое PROFIBUS устройство DPSlave_12 с одним извлеченным модулем	Пример 3:Ведомое PROFIBUS устройство DPSlave_12 отключено
1: Конфигурация устройства активна	0x01FC_FF03	0x01FC_FF03	0x01FC_FF03
2: Устройство неисправно	0x0000_0000	0x0110_0000	0x0110_0000
3: Устройство деактивировано	0x0000_0000	0x0000_0000	0x0000_0000
4: Устройство существует	0x01FC_FF03	0x01FC_FF03	0x01EC_FF03
5: Проблема в устройстве	0x0000_0000	0x0110_0000	0x0110_0000

В следующих четырех таблицах показана двоичная разбивка четырех анализируемых байтов данных:

Таблица 9-159 Пример 1: Ошибки отсутствуют: Для MODE 1 (конфигурация устройства активна) возвращается значение 0x01FC_FF03.

Байт со значением	Битовая комбинация со значением	Примечания
Байт 1 0x01	Бит 7 0000-0001 Бит 0	Бит 0 = TRUE; данные доступны.
Байт 2 0xFC	Бит 15 1111-1100 Бит 8	
Байт 3 0xFF	Бит 23 1111-1111 Бит 16	
Байт 4 0x03	Бит 31 0000-0011 Бит 24	

Устройства сконфигурированы по адресам от 10 (бит 10) до 25 (бит 25).
По адресам от 1 до 9 нет сконфигурированных устройств.Данные
MODE 4 (устройство существует) совпадают с MODE 1 (конфигурация устройства активна), т.е. сконфигурированные устройства соответствуют имеющимся устройствам.

Таблица 9-160 Пример 2: Модуль был извлечен из ведомого PROFIBUS устройства "DPSlave_12". Для MODE 2 (устройство неисправно) возвращается значение 0x0110_0000.

Байт со значением	Битовая комбинация со значением	Примечания
Байт 1 0x01	Бит 7 0000-0001 Бит 0	Бит 0 = TRUE; данные доступны.
Байт 2 0x10	Бит 15 000**1**-0000 Бит 8	
Байт 3 0x00	Бит 23 0000-0000 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Устройство 12 (бит 12) обозначено как неисправное.
MODE 5 (проблема в устройстве) возвращает ту же информацию, что и MODE 2 (устройство неисправно).

Таблица 9-161 Пример 2 (продолжение): Модуль был извлечен из ведомого PROFIBUS устройства "DPSlave_12". Для MODE 4 (устройство существует) возвращается значение 0x01FC_FF03.

Байт со значением	Битовая комбинация со значением	Примечания
Байт 1 0x01	Бит 7 0000-0001 Бит 0	Бит 0 = TRUE; данные доступны.
Байт 2 0xFC	Бит 15 111**1**-1100 Бит 8	
Байт 3 0xFF	Бит 23 1111-1111 Бит 16	
Байт 4 0x03	Бит 31 0000-0011 Бит 24	

Хотя устройство 12 (бит 12), как показано выше в MODE 2 (устройство неисправно), имеет ошибку, устройство продолжает работать в сети, поэтому MODE 4 (устройство существует) отображает устройство как "существующее устройство".

Таблица 9-162 Пример 3: Ведомое PROFIBUS устройство "DPSlave_12" отсоединено от сети PROFIBUS (кабель отключен или потеря напряжения). "DPSlave_12" продолжает опознаваться как неисправное устройство и как устройство с ошибкой. Разница в том, что "DPSlave_12" больше не определяется как существующее устройство. Для MODE 4 (устройство существует) возвращается значение 0x01EC_FF03.

Байт со значением	Битовая комбинация со значением	Примечания
Байт 1 0x01	Бит 7 0000-0001 Бит 0	Бит 0 = TRUE; данные доступны.
Байт 2 0xEC	Бит 15 111**0**-1100 Бит 8	
Байт 3 0xFF	Бит 23 1111-1111 Бит 16	
Байт 4 0x03	Бит 31 0000-0011 Бит 24	

Устройство 12 (бит 12) обозначено как несуществующее. За этим исключением, остальные устройства с 10 по 25 продолжают отображаться как существующие.

Пример в PROFINET

Пример в PROFINET состоит из следующих компонентов:

- 16 ведомых PROFINET устройств с именами "et200s_1" - "et200s_16".
- 16 устройств PROFINET используют по одному из номеров PROFINET устройств от 1 до 16 каждый.
- Для каждого ведомого устройства сконфигурировано несколько модулей ввода-вывода.
- Отображаются первые четыре байта выводимой информации параметра STATE.

MODE	Пример 1: Штатный режим работы без ошибок	Пример 2:Модуль ведомого PROFINET устройства et200s_1 извлечен	Пример 3:Ведомое PROFINET устройство et200s_1 отсоединено
1: Конфигурация устройства активна	0xFFFF_0100	0xFFFF_0100	0xFFFF_0100
2 - Устройство неисправно	0x0000_0000	0x0300_0000	0x0300_0000

MODE	Пример 1: Штатный режим работы без ошибок	Пример 2:Модуль ведомого PROFINET устройства et200s_1 извлечен	Пример 3:Ведомое PROFINET устройство et200s_1 отсоединено
3 - Устройство деактивировано	0x0000_0000	0x0000_0000	0x0000_0000
4 - Устройство существует	0xFFFF_0100	0xFFFF_0100	0xFDFF_0100
5 - Проблема в устройстве	0x0000_0000	0x0300_0000	0x0300_0000

В следующих четырех таблицах показана двоичная разбивка четырех анализируемых байтов данных:

Таблица 9-163 Пример 1: Ошибки отсутствуют: Для MODE 1 (конфигурация устройства активна) возвращается значение 0xFFFF_0100.

Байт со значением	Битовая комбинация со значением	Примечания
Байт 1 0xFF	Бит 7 1111-1111 Бит 0	Бит 0 = TRUE; данные доступны.
Байт 2 0xFF	Бит 15 1111-1111 Бит 8	
Байт 3 0x01	Бит 23 0000-0001 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Устройства сконфигурированы по адресам от 1 (бит 1) до 16 (бит 16).
По адресам от 1 до 9 нет сконфигурированных устройств.Данные
MODE 4 (устройство существует) совпадают с MODE 1 (конфигурация устройства активна), т.е. сконфигурированные устройства соответствуют имеющимся устройствам.

Таблица 9-164 Пример 2: Модуль был извлечен из ведомого PROFINET устройства "et200s_1". Для MODE 2 (устройство неисправно) возвращается значение 0x0300_0000.

Байт со значением	Битовая комбинация со значением	Примечания
Байт 1 0x03	Бит 7 0000-00**1**1 Бит 0	Бит 0 = TRUE; данные доступны.
Байт 2 0x00	Бит 15 0000-0000 Бит 8	
Байт 3 0x00	Бит 23 0000-0000 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Устройство 1 (бит 1) обозначено как неисправное. Т.к. устройство существует как и прежде, MODE 4 (устройство существует) показывает те же данные, что и в штатном режиме.
MODE 5 (проблема в устройстве) возвращает ту же информацию, что и MODE 2 (устройство неисправно).

Таблица 9-165 Пример 2 (продолжение): Модуль был извлечен из ведомого PROFIBUS устройства "et200s_1". Для MODE 4 (устройство существует) возвращается значение 0xFFFF_0100.

Байт со значением	Битовая комбинация со значением	Примечания
Байт 1 0xFF	Бит 7 1111-11**1**1 Бит 0	Бит 0 = TRUE; данные доступны.
Байт 2 0xFF	Бит 15 1111-1111 Бит 8	
Байт 3 0x01	Бит 23 0000-0001 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Хотя устройство 1 (бит 1), как показано выше в MODE 2 (устройство неисправно), имеет ошибку, устройство продолжает работать в сети, поэтому MODE 4 (устройство существует) отображает устройство как "существующее устройство".

Таблица 9-166 Пример 3: Ведомое PROFINET устройство "et200s_1" отсоединено от сети PROFINET (кабель отключен или потеря напряжения). Для MODE 4 (устройство существует) возвращается значение 0xFDFF_0100.

Байт со значением	Битовая комбинация со значением	Примечания
Байт 1 0xFD	Бит 7 1111-11**0**1 Бит 0	Бит 0 = TRUE; данные доступны.
Байт 2 0xFF	Бит 15 1111-1111 Бит 8	
Байт 3 0x01	Бит 23 0000-0001 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Устройство 1 (бит 1) не существует. Устройства 2 (бит 2) до 16 (бит 16) существуют.

9.7.8 Инструкция ModuleStates

С помощью инструкции возможен вывод состояния всех модулей на станции PROFIBUS или PROFINET.

Таблица 9-167 Инструкция ModuleStates

LAD/FBD	SCL	Описание

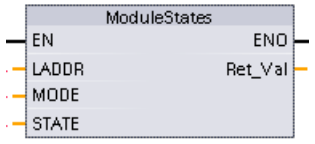
	`ret_val := ModuleStates(`
	`laddr:=_word_in_,`
	`mode:=_uint_in,`
	`state:=_variant_inout);`
		ModuleStates опрашивает рабочие состояния модулей ввода-вывода. После выполнения параметр STATE содержит состояние ошибки каждого отдельного модуля ввода/вывода в списке битов (для назначенных параметров LADDR и MODE). Эта информация соответствует состоянию модуля в окне диагностики STEP 7.
		Вход LADDR инструкции ModuleStates использует аппаратный идентификатор станции распределенного ввода-вывода, а не головного модуля. Для определения аппаратного идентификатора выбрать всю станцию на вкладке сети и после посмотреть информацию в свойствах в области "Аппаратный идентификатор". Также он может быть найден через типы данных "Hw_Device" и "Hw_DpSlave" на вкладке системных констант в таблиц переменных (тегов) PLC.

Таблица 9-168 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
LADDR	IN	HW_DEVICE	Логический адрес (идентификация модулей ввода-вывода)
MODE	IN	UInt	Поддерживает пять режимов работы. Вход MODE устанавливает, какие данные будут выводиться в указанном для информации STATE месте. Возможны следующие режимы работы: • 1: Конфигурация модуля активна • 2: Модуль неисправен • 3: Модуль деактивирован • 4: Модуль существует • 5: Проблема в модуле

Параметр и тип данных		Тип данных	Описание
RET_VAL	OUT	Int	Статус (код условия)
STATE[1]	InOut	Variant	Буфер, принимающий состояние ошибки от каждого модуля: В качестве типа данных, для параметра STATE может быть выбран любой тип битовых данных (Bool, Byte, Word или DWord) или массив битовых данных. • Бит 0 первого байта выводимых данных STATE это сводный бит. Если он установлен на TRUE, то это означает, что доступны и другие данные. • Выводимые параметром STATE данные показывают полное соответствие между позицией бита и позицией модуля. Эта адресация слота TRUE для PROFIBUS и PROFINET. Пример: У ET 200SP с головным модулем, силовым модулем и парой модулей ввода-вывода, Бит 1 в первом байте соответствует головному модулю, Бит 2 - силовому модулю, а Бит 3 и 4 - модулям ввода-вывода.

[1] Может быть назначено до 128 бит. Число необходимых бит зависит от цели использования модуля ввода-вывода.

Таблица 9-169 Коды условий

RET_VAL (W#16#...)	Описание
0	Ошибки отсутствуют
8091	Указанный LADDR модуль отсутствует
8092	Указанный LADDR модуль не обращается к устройству ввода-вывода.
8093	Недействительный тип данных для параметра STATE: Действительные типы данных (Bool, Byte, Word или Dword) или массив из (Bool, Byte, Word или Dword).
80Bx	Инструкция ModuleStates не поддерживается этим CPU для этого параметра LADDR.
8452	Полные данные о состоянии являются слишком большими для назначенного параметра STATE . Буфер STATE содержит частичный результат.

9.7.8.1 Примеры для конфигурации ModuleStates

Пример в PROFIBUS

Пример в PROFIBUS состоит из следующих компонентов:

- 16 PROFIBUS-устройств с именами "DPSlave_10" до "DPSlave_25"
- 16 PROFIBUS-устройств используют по одному из PROFIBUS-адресов от 10 до 25 каждый.
- Для каждого ведомого устройства сконфигурировано несколько модулей ввода-вывода.
- В примере используется параметр LADDR ведомого PROFIBUS устройства "DPSlave_12", содержащего головной модуль, силовой модуль и два модуля ввода-вывода.
- Отображаются первые четыре байта выводимой информации параметра STATE.

MODE	Пример 1: Штатный режим работы без ошибок	Пример 2:Ведомое PROFIBUS устройство DPSlave_12 с извлеченным модулем	Пример 3:Ведомое PROFIBUS устройство DPSlave_12 отключено
1: Конфигурация модуля активна	0x1F00_0000	0x1F00_0000	0x1F00_0000
2: Модуль неисправен	0x0000_0000	0x0900_0000	0x1F00_0000
3: Модуль деактивирован	0x0000_0000	0x0000_0000	0x0000_0000
4: Модуль существует	0x1F00_0000	0x1700_0000	0x0000_0000
5: Проблема в модуле	0x0000_0000	0x0900_0000	0x1F00_0000

В следующих четырех таблицах показана двоичная разбивка четырех анализируемых байтов данных:

Таблица 9-170 Пример 1: Ошибки отсутствуют: Для MODE 1 (конфигурация модуля активна) возвращается значение 0x1F00_0000.

Байт со значением	Битовая комбинация со значением	Примечания
Байт 1 0x1F	Бит 7 0001-1111 Бит 0	Бит 0 = TRUE; данные доступны.
Байт 2 0x00	Бит 15 0000-0000 Бит 8	
Байт 3 0x00	Бит 23 0000-0000 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Слоты 1 (бит 1) до 4 (бит 4) заняты модулями. Слот 5 (бит 5) и прочие не заняты модулями. Данные MODE 4 (модуль существует) совпадают с MODE 1 (конфигурация модуля активна), т.е. сконфигурированные модули соответствуют имеющимся модулям.

Таблица 9-171 Пример 2: Модуль был извлечен из ведомого PROFIBUS устройства "DPSlave_12". Для MODE 2 (модуль неисправен) возвращается значение 0x0900_0000.

Байт со значением	Битовая комбинация со значением	Примечания
Байт 1 0x09	Бит 7 0000-**1**001 Бит 0	Бит 0 = TRUE; данные доступны.
Байт 2 0x00	Бит 15 0000-0000 Бит 8	
Байт 3 0x00	Бит 23 0000-0000 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Только модуль 3 (бит 3) обозначен как неисправный. Все другие модули находятся в работоспособном состоянии.

Таблица 9-172 Пример 2 (продолжение): Модуль был извлечен из ведомого PROFIBUS устройства "DPSlave_12". Для MODE 4 (модуль неисправен) возвращается значение 0x1700_0000.

Байт со значением	Битовая комбинация со значением	Примечания
Байт 1 0x17	Бит 7 0001-**0**111 Бит 0	Бит 0 = TRUE; данные доступны.
Байт 2 0x00	Бит 15 0000-0000 Бит 8	
Байт 3 0x00	Бит 23 0000-0000 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Модуль 3 (бит 3) отображается как отсутствующий. Модули 1, 2 и 4 (биты 1, 2 и 4) отображаются как существующие.

Таблица 9-173 Пример 3: Ведомое PROFIBUS устройство "DPSlave_12" отсоединено от сети PROFIBUS (кабель отключен или потеря напряжения). Для MODE 2 (модуль неисправен) возвращается значение 0x1F00_0000.

Байт со значением	Битовая комбинация со значением	Примечания
Байт 1 0x1F	Бит 7 000**1**-1111 Бит 0	Бит 0 = TRUE; данные доступны.
Байт 2 0x00	Бит 15 0000-0000 Бит 8	
Байт 3 0x00	Бит 23 0000-0000 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Все модули в слотах от 1 о 4 (биты 1 до 4) обозначаются как неисправные, так как устройство отсутствует.
MODE 5 (проблема в модуле) отображает ту же информацию, что и MODE 2 (устройство неисправно).

Пример в PROFINET

Пример в PROFINET состоит из следующих компонентов:

- 16 ведомых PROFINET устройств с именами "et200s_1" - "et200s_16".
- 16 устройств PROFINET используют по одному из номеров PROFINET устройств от 1 до 16 каждый.
- Для каждого ведомого устройства сконфигурировано несколько модулей ввода-вывода.
- В примере используется ведомое PROFINET устройство "et200s_1", содержащее головной модуль, силовой модуль и 18 модулей ввода-вывода.
- Отображаются первые четыре байта выводимой информации параметра STATE.

MODE	Пример 1: Штатный режим работы без ошибок	Пример 2:Модуль ведомого PROFINET устройства et200s_1 извлечен	Пример 3:Ведомое PROFINET устройство et200s_1 отсоединено
1: Конфигурация модуля активна	0xFFFF_1F00	0xFFFF_1F00	0xFFFF_1F00
2: Модуль неисправен	0x0000_0000	0x0180_0000	0xFFFF_1F00
3: Модуль деактивирован	0x0000_0000	0x0000_0000	0x0000_0000
4: Модуль существует	0xFFFF_1F00	0xFF7F_1F00	0x0000_0000
5: Проблема в модуле	0x0000_0000	0x0180_0000	0xFFFF_1F00

В следующих четырех таблицах показана двоичная разбивка четырех анализируемых байтов данных:

Таблица 9-174 Пример 1: Ошибки отсутствуют: Для MODE 1 (конфигурация модуля активна) возвращается значение 0xFFFF_1F00.

Байт со значением	Битовая комбинация со значением	Примечания
Байт 1 0xFF	Бит 7 1111-1111 Бит 0	Бит 0 = TRUE; данные доступны.
Байт 2 0xFF	Бит 15 1111-1111 Бит 8	
Байт 3 0x1F	Бит 23 0001-1111 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Слоты 1 (бит 1) до 20 (бит 20) заняты модулями. Слот 21 (бит 21) и прочие не заняты модулями.
Данные
MODE 4 (модуль существует) совпадают с MODE 1 (конфигурация модуля активна), т.е. сконфигурированные модули соответствуют имеющимся модулям.

Таблица 9-175 Пример 2: Модуль был извлечен из ведомого PROFINET устройства "et200s_1". Для MODE 2 (модуль неисправен) возвращается значение 0x0180_0000.

Байт со значением	Битовая комбинация со значением	Примечания
Байт 1 0x01	Бит 7 0000-0001 Бит 0	Бит 0 = TRUE; данные доступны.
Байт 2 0x80	Бит 15 **1**000-0000 Бит 8	
Байт 3 0x00	Бит 23 0000-0000 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Только модуль 15 (бит 15) обозначен как неисправный. Все другие модули находятся в работоспособном состоянии.

Таблица 9-176 Пример 2 (продолжение): Модуль был извлечен из ведомого PROFIBUS устройства "et200s_1". Для MODE 4 (модуль неисправен) возвращается значение 0xFF7F_1F00.

Байт со значением	Битовая комбинация со значением	Примечания
Байт 1 0xFF	Бит 7 1111-1111 Бит 0	Бит 0 = TRUE; данные доступны.
Байт 2 0x7F	Бит 15 **0**111-1111 Бит 8	
Байт 3 0x1F	Бит 23 0001-1111 Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Модуль 15 (бит 15) отображается как отсутствующий. Модули от 1 до 14 (биты 1 - 14) и от 16 до 20 (биты 16 - 20) отображаются как отсутствующие.

Таблица 9-177 Пример 3: Ведомое PROFINET устройство "et200s_1" отсоединено от сети PROFINET (кабель отключен или потеря напряжения). Для MODE 2 (модуль неисправен) возвращается значение 0xFFFF_1F00.

Байт со значением	Битовая комбинация со значением	Примечания
Байт 1 0xFF	Бит 7 **1111-1111** Бит 0	Бит 0 = TRUE; данные доступны.
Байт 2 0xFF	Бит 15 **1111-1111** Бит 8	
Байт 3 0x1F	Бит 23 000**1**-**1111** Бит 16	
Байт 4 0x00	Бит 31 0000-0000 Бит 24	

Все модули в слотах от 1 до 20 (биты 1 до 20) обозначаются как неисправные, так как устройство отсутствует.
MODE 5 (проблема в модуле) отображает ту же информацию, что и MODE 2 (устройство неисправно).

9.7.9 GET_DIAG (чтение информации системы диагностики)

Описание

С помощью инструкции GET_DIAG считывается информация системы диагностики аппаратного устройства. Аппаратное устройство выбирается параметром LADDR. С помощью параметра MODE выбирается, какая информация системы диагностики должна быть считана.

Таблица 9-178 Инструкция GET_DIAG

LAD/FBD	SCL	Описание

	`ret_val := GET_DIAG(`
	`  mode:=_uint_in_,`
	`  laddr:=_word_in_,`
	`  cnt_diag=>_uint_out_,`
	`  diag:=_variant_inout_,`
	`  detail:=_variant_inout_);`
		Считывает информацию системы диагностики из назначенного аппаратного устройства

Параметр

В приведенной ниже таблице показаны параметры инструкции GET_DIAG:

Таблица 9-179 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
MODE	IN	UInt	С помощью параметра MODE указывается, какие диагностические данные должны возвращаться.
LADDR	IN	HW_ANY (Word)	Аппаратный идентификатор устройства
RET_VAL	OUT	Int	Состояние инструкции
CNT_DIAG	OUT	UInt	Количество выведенных сведений диагностики
DIAG	InOut	Variant	Указатель на область данных для сохранения информации системы диагностики выбранного режима
DETAILS	InOut	Variant	Указатель на область данных для сохранения информации системы диагностики в соответствии с выбранным режимом

Параметр MODE

В зависимости от значения параметра MODE, различные диагностические данные возвращаются в выходных параметрах DIAG, CNT_DIAG и DETAILS:

Таблица 9-180 Параметр MODE

MODE	Описание	DIAG	CNT_DIAG	DETAILS
0	Вывод всей поддерживаемой информации системы диагностики для модуля в формате DWord, где бит X=1 показывает, что поддерживается режим X.	Битовая строка поддерживаемых режимов в формате DWord, где бит X=1 показывает, что поддерживается режим X. Если параметр MODE = 0, то S7-1200 CPU игнорирует параметр LADDR.	0	-
1	Вывод соответствующего состояния адресованного аппаратного объекта.	Диагностическое состояние: Вывод согласно структуре DIS. (Примечание: См. также информацию о "DIS структуре" ниже и пример для инструкции GET_DIAG в конце раздела.)	0	-
2	Вывод состояния всех подчиненных модулей адресованного аппаратного объекта.	Вывод диагностических данных согласно структуре DNN. (Примечание: См. также информацию о "DNN структуре" ниже и пример для инструкции GET_DIAG в конце раздела.)	0	-

Структура DIS

Если параметр MODE = 1, то информация системы диагностики выводится в соответствии со структурой DIS. В следующей таблице объясняется значение отдельных параметров:

Таблица 9-181 Структура источника информации системы диагностики (DIS, Diagnostic Information Source)

Параметр	Тип данных	Значение	Описание
MaintenanceState	DWord	Enum	
		0	Обслуживание не требуются
		1	Модуль или устройство деактивированы.
		2	-
		3	-
		4	-
		5	Требуется техобслуживание
		6	Запрос на техобслуживание
		7	Ошибка
		8	Неизвестное состояние/ошибка в подчиненном модуле
		9	-
		10	Входы/выходы недоступны.

Параметр	Тип данных	Значение	Описание
Componentstate Detail	DWord	Битовый массив	Состояние субмодулей модуля: • Бит от 0 до 15: Сообщение о состоянии модуля • Бит от 16 до 31: Сообщение о состоянии CPU
		От 0 до 2 (Enum)	Дополнительная информация: • Бит 0: Не содержит дополнительной информации • Бит 1: Передача не разрешена
		3	Бит 3 = 1: Как минимум один канал поддерживает квалификаторы для диагностики.
		4	Бит 4 = 1: Как минимум один канал или компонент нуждается в техническом обслуживании
		5	Бит 5 = 1: Как минимум для одного канала или компонента было запрошено техобслуживание
		6	Бит 6 = 1: Ошибка как минимум в одном канале или компоненте
		От 7 до 10	Зарезервировано (всегда = 0)
		От 11 до 14	Бит 11 = 1: PNIO - правильный субмодуль Бит 12 = 1: PNIO - запасной модуль Бит 13 = 1: PNIO - неправильный модуль Бит 14 = 1: PNIO - модуль отсоединен
		15	Зарезервировано (всегда = 0)
		От 16 до 31	Созданная CPU дополнительная информация для модулей: Бит 16 = 1: Модуль деактивирован Бит 17 = 1: Активна операция CiR Бит 18 = 1: Вход недоступен Бит 19 = 1: Выход недоступен Бит 20 = 1: Переполнение буфера диагностики Бит 21 = 1: Диагностика недоступна Бит 22 - 31: Зарезервировано (всегда 0)
OwnState	Uint16	Enum	Значение параметра OwnState описывает состояние техобслуживания модуля.
		0	Неполадки отсутствуют
		1	Модуль или устройство деактивированы.
		2	Требуется техобслуживание
		3	Запрос на техобслуживание
		4	Ошибка
		5	Модуль или устройство недоступны для CPU (для модулей и устройств, относящихся в одному CPU).
		6	Входы/выходы недоступны.
		7	-

Параметр	Тип данных	Значение	Описание
IO State	Uint16	Битовый массив	Состояние ввода / вывода модуля
		0	Бит 0 = 1: Обслуживание не требуются
		1	Бит 1 = 1: Модуль или устройство деактивированы.
		2	Бит 2 = 1: Требуется техобслуживание
		3	Бит 3 = 1: Запрос на техобслуживание
		4	Бит 4 = 1: Ошибка
		5	Бит 5 = 1: Модуль или устройство недоступны для CPU (для модулей и устройств, относящихся в одному CPU).
		6	Квалификатор; бит 7 = 1, если установлен бит 0, 2 или 3
		7	Входы/выходы недоступны.
		От 8 до 15	Зарезервировано (всегда = 0)
OperatingState	UInt16	Enum	
		0	-
		1	В режиме STOP/обновление прошивки
		2	В режиме STOP/восстановление исходного состояния памяти
		3	В режиме STOP/самопуск
		4	В режиме STOP
		5	Восстановление исходного состояния памяти
		6	В режиме START
		7	В режиме RUN
		8	-
		9	В режиме HOLD
		10	-
		11	-
		12	Модуль неисправен
		13	-
		14	Нет напряжения
		15	CiR
		16	В режиме STOP/без DIS
		17	Вход
		18	
		19	
		20	

Структура DNN

Если параметр MODE = 2, то информация системы диагностики выводится в соответствии со структурой DNN. В следующей таблице объясняется значение отдельных параметров:

Таблица 9-182 Структура диагностического навигационного узла (DNN, Diagnostic Navigation Node)

Параметр	Тип данных	Значение	Описание
SubordinateState	UINT	Enum	Состояние подчиненного модуля (см. параметр OwnState структуры DIS)
SubordinateIOState	WORD	Bitarray	Состояние входов и выходов подчиненного модуля (см. параметр IO State структуры DIS)
DNNmode	WORD	Bitarray	• Бит 0 = 0: Диагностика активирована • Бит 0 = 1: Диагностика деактивирована • Бит от 1 до 15: Зарезервировано

Параметр RET_VAL

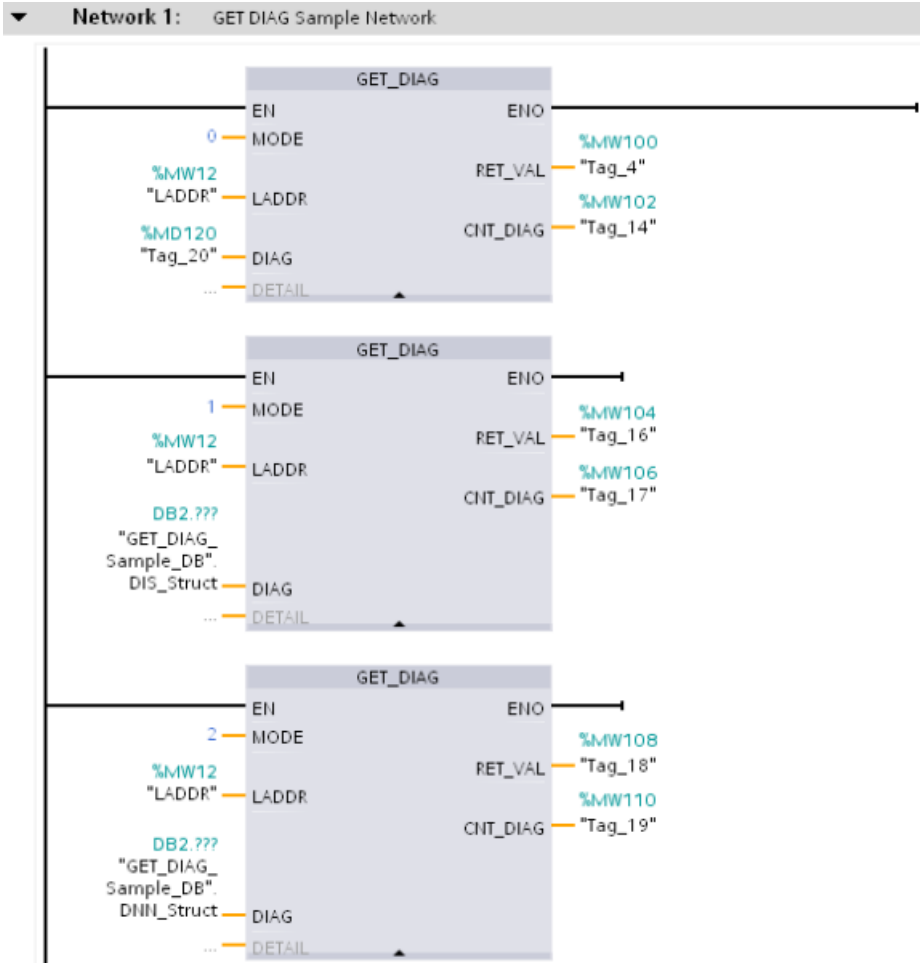
Таблица 9-183 Коды ошибок параметра RET_VAL

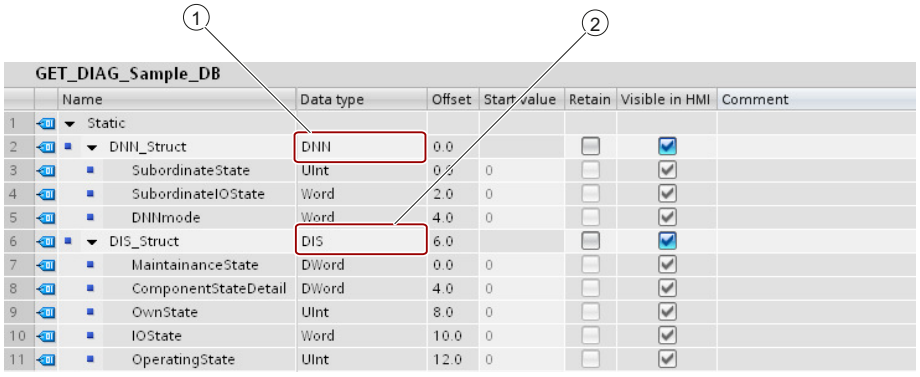
Код ошибки (W#16#...)	Описание
0	Ошибки отсутствуют
8080	Значение в параметре MODE не поддерживается.
8081	Тип в параметре DIAG не поддерживается выбранным модулем (параметр MODE).
8082	Тип в параметре DETAILS не поддерживается выбранным модулем (параметр MODE).
8090	LADDR отсутствует.
8091	Выбранный канал в параметре CHANNEL отсутствует.
80C1	Недостаточно ресурсов для параллельной работы

Пример

Следующий сегмент PKC и следующий DB показывают, как использовать три режима работы с тремя структурами:

- DIS
- DNN





① DNN

② DIS

Примечание

В DB тип данных должен быть введен вручную, чтобы обращаться к каждой из трех структур. Выбор через выпадающий список невозможен. Ввести типы данных точно так, как показано ниже:

- DNN
- DIS

9.7.10 Диагностические события для распределенной периферии

Примечание

В системе PROFIBUS IO модуль CPU после процесса загрузки или выключения и повторного включения переходит в рабочее состояние RUN, если только не используется установка аппаратной совместимости, при которой разрешены допустимые запасные модули (Страница 167) и отсутствует минимум один модуль или при этом речь идет не о допустимом запасном модуле для сконфигурированного модуля.

Как видно из следующей таблицы, CPU поддерживает такую диагностику, которая может быть сконфигурирована для компонентов системы распределенного ввода-вывода. Каждая из этих ошибок создает запись в журнал буфера диагностики.

Таблица 9-184 Обработка диагностических событий для PROFINET и PROFIBUS

Тип ошибки	Информация системы диагностики для станции?	Запись в буфер диагностики?	Рабочее состояние CPU
Диагностическая ошибка	Да	Да	Остается в рабочем состоянии RUN
Ошибка стойки или станции	Да	Да	Остается в рабочем состоянии RUN
Ошибка доступа к периферии [1]	Нет	Да	Остается в рабочем состоянии RUN
Ошибка доступа к периферии [2]	Нет	Да	Остается в рабочем состоянии RUN
Событие извлечения/вставки	Да	Да	Остается в рабочем состоянии RUN

[1] Пример причины для ошибки доступа к периферии: Модуль был удален.

[2] Пример причины для ошибки доступа к периферии: Ациклическая коммуникация с модулем, не выполняющим коммуникацию.

С помощью инструкции GET_DIAG (Страница 479) можно запрашивать информацию системы диагностики для любой станции. Таким образом можно обрабатывать возникающие в устройстве ошибки на программном уровне и, при желании, переводить CPU в рабочее состояние STOP. Для этого необходимо указать аппаратное устройство, из которого должна считываться информация о состоянии.

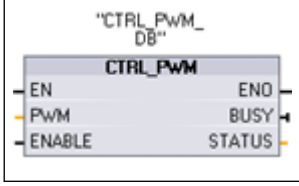
Инструкция GET_DIAG использует "L-адрес" (LADDR) станции для запросы состояния всей станции. Этот L-адрес можно найти на вкладке сети конфигурации устройства, выбрав всю стойку станции (всю серую область). L-адрес будет отображаться на вкладке "Свойства" станции. Параметр LADDR для каждого отдельного модуля можно найти либо в свойствах модуля (в конфигурации устройства) или в стандартной таблице переменных (тегов) модуля CPU.

9.8 Импульс

9.8.1 CTRL_PWM (широтно-импульсная модуляция)

Таблица 9-185 Инструкция CTRL_PWM (широтно-импульсная модуляция)

LAD/FBD	SCL	Описание

	`"CTRL_PWM_DB"(`
	`PWM:=_uint_in_,`
	`ENABLE:=_bool_in_,`
	`BUSY=>_bool_out_,`
	`STATUS=>_word_out_);`
		Предлагает фиксированное время цикла с переменной относительной продолжительностью включения. Выход PWM после запуска выполняет последовательные операции с указанной частотой (время цикла). Ширина импульса изменяется так, как требуется, чтобы достичь необходимого эффекта.

[1] При вставке инструкции STEP 7 открывает диалоговое окно"Опции вызова" для создания соответствующего DB.

[2] В примере SCL "CTRL_PWM_DB" - это имя DB экземпляра.

Таблица 9-186 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
PWM	IN	HW_PWM (Word)	Идентификация PWM: Имена активированных генераторов импульсов становятся переменными (тегами) в таблице переменных (тегов) "Константы" и могут использоваться в качестве параметров PWM. (значение по умолчанию: 0)
ENABLE	IN	Bool	1 = запустить генератор импульсов 0 = остановить генератор импульсов
BUSY	OUT	Bool	Функция повреждения (значение по умолчанию: 0)
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

Инструкция CTRL_PWM сохраняет информацию параметра в DB. Параметры блока данных изменяются не отдельно пользователем, а под управлением инструкции CTRL_PWM.

Указать требуемый генератор импульсов через имя переменной (тега) для параметра PWM.

Если вход EN имеет значение TRUE, то инструкция PWM_CTRL запускает и останавливает указанную PWM на основе значения на входе ENABLE. Длительность импульса задается значением назначенного адреса выходного слова.

Т.к. CPU обрабатывает запрос при выполнении инструкции CTRL_PWM, параметр BUSY всегда возвращает FALSE. При обнаружении ошибки, ENO устанавливается на FALSE и параметр STATUS содержит код ошибки.

Длительность импульса при первом переходе CPU в режиме RUN устанавливается на введенное в конфигурации устройства начальное значение. Для изменения длительности импульса записать требуемые значения на указанный в конфигурации устройства адрес выходного слова ("Выходные адреса"/"Начальный адрес:"). Для записи требуемой длительности импульса в соответствующее выходное слово, следует использовать инструкцию, например, для передачи, преобразования, арифметических операций или ПИД. Необходимо соблюдать действительный диапазон для значения выходного слова (процент, тысячная часть, десятитысячная часть или аналоговый формат S7).

Примечание

Принудительная активация назначенных PWM и PTO цифровых I/O невозможна.

Используемые широтно-импульсной модуляцией (PWM) и выводом последовательности импульсов (PTO) цифровые I/O назначаются при конфигурировании устройства. Если этим функциям назначаются цифровые I/O, то значения адресов назначенных I/O не могут быть изменены с помощью функции принудительного присваивания значений в таблице текущего контроля.

Таблица 9-187 Значение параметра STATUS

STATUS	Описание
0	Ошибки отсутствуют
80A1	Идентификатор PWM не адресует действительную PWM.

9.8.2 CTRL_PTO (вывод последовательности импульсов)

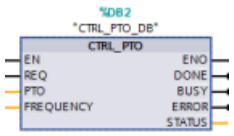
Инструкция PTO предоставляет выход прямоугольных импульсов с относительной продолжительностью включения в 50 % с указанной частотой. С помощью инструкции CTRL_PTO возможно назначение частоты без технологического объекта (TO).

Для этой инструкции нужен генератор импульсов. Следует активировать генератор импульсов в конфигурации оборудования и выбрать тип сигнала. Дополнительную информацию можно найти здесь "Конфигурирование импульсного канала для PWM или PTO" (Страница 494).

Доступ к инструкции CTRL_PTO возможен через окно задач, расширенные инструкции.

Таблица 9-188 Инструкция CTRL_PTO (вывод последовательности импульсов)

LAD/FBD [1]	SCL [2]	Описание

	`"CTRL_PTO_DB"(`
	`REQ:=_bool_in_,`
	`PTO:=_uint_in_,`
	`FREQUENCY:=_udint_in_,`
	`DONE=>_bool_out_,`
	`BUSY=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		Инструкция PTO позволяет управлять частотой выхода прямоугольных импульсов (с относительной продолжительностью включения в 50 %).

[1] При вставке инструкции STEP 7 открывает диалоговое окно "Опции вызова" для создания соответствующего DB.

[2] В примере SCL "CTRL_PTO_DB" - это имя DB экземпляра.

Таблица 9-189 Типы данных для параметров

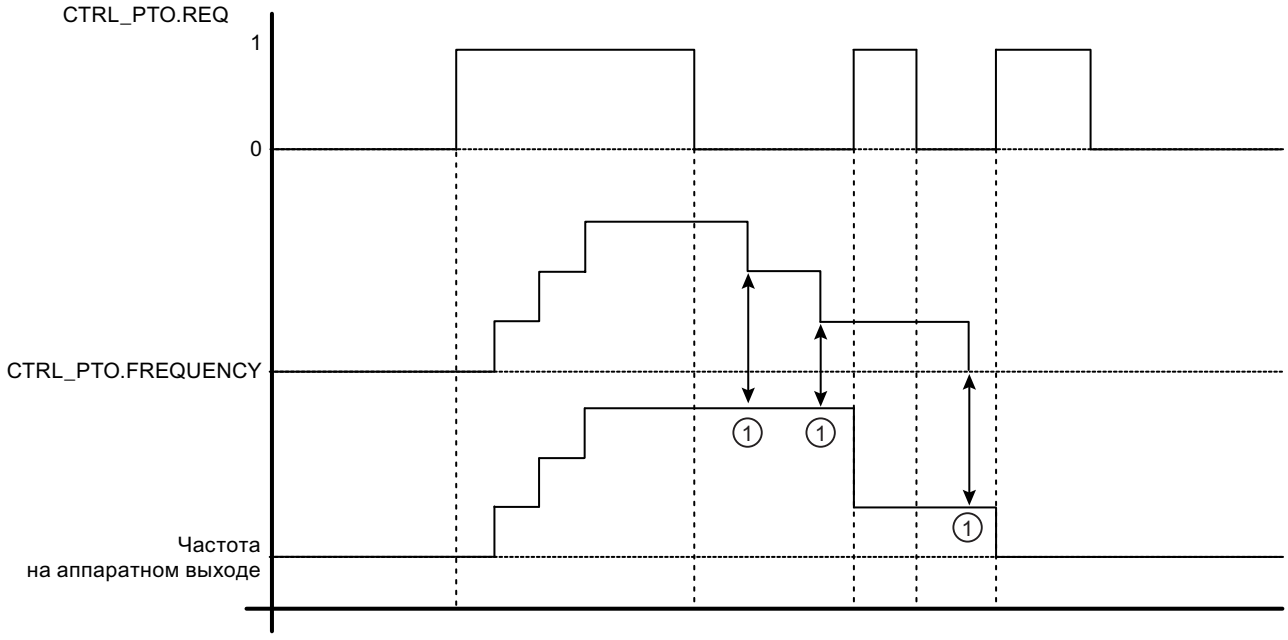
Параметр и тип данных		Тип данных	Описание
EN	IN	Bool	1 = инструкция активирована 0 = инструкция не активировано
REQ	IN	Bool	1 = установить выходную частоту PTO на значение входа FREQUENCY 0 = PTO без изменений
PTO	IN	HW_PTO (Word)	Идентификатор PTO: Аппаратный идентификатор генератора импульсов: • Имена активированных генераторов импульсов становятся переменными (тегами) в таблице переменных (тегов) "Константы" и могут использоваться в качестве параметров PTO. (по умолчанию = 0.) • ID оборудования можно узнать из свойств генератора импульсов на вкладке оборудования. Системные константы также содержат ID оборудования для генераторов импульсов. (по умолчанию = 0.)
FREQUENCY	IN	UDInt	Требуемая частота PTO (в Гц). Это значение действует только при REQ = 1 (значение по умолчанию 0 Гц).
DONE	OUT	Bool	Функция завершена без ошибок (значение по умолчанию: 0)
BUSY	OUT	Bool	Функция повреждения (значение по умолчанию: 0)
ERROR	OUT	Word	Обнаружена ошибка (значение по умолчанию: 0)
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

Инструкция CTRL_PTO сохраняет информацию параметра в DB. Параметры блока данных изменяются не отдельно пользователем, а под управлением инструкции CTRL_PTO.

Указать требуемый генератор импульсов через имя переменной (тега) или идентификатор оборудования в параметре PTO.

Если вход EN имеет значение TRUE, то инструкция CTRL_PTO запускает и останавливает указанный PTO. Если вход EN = FALSE, то инструкция CTRL_PTO не выполняется и PTO остается в своем текущем состоянии.

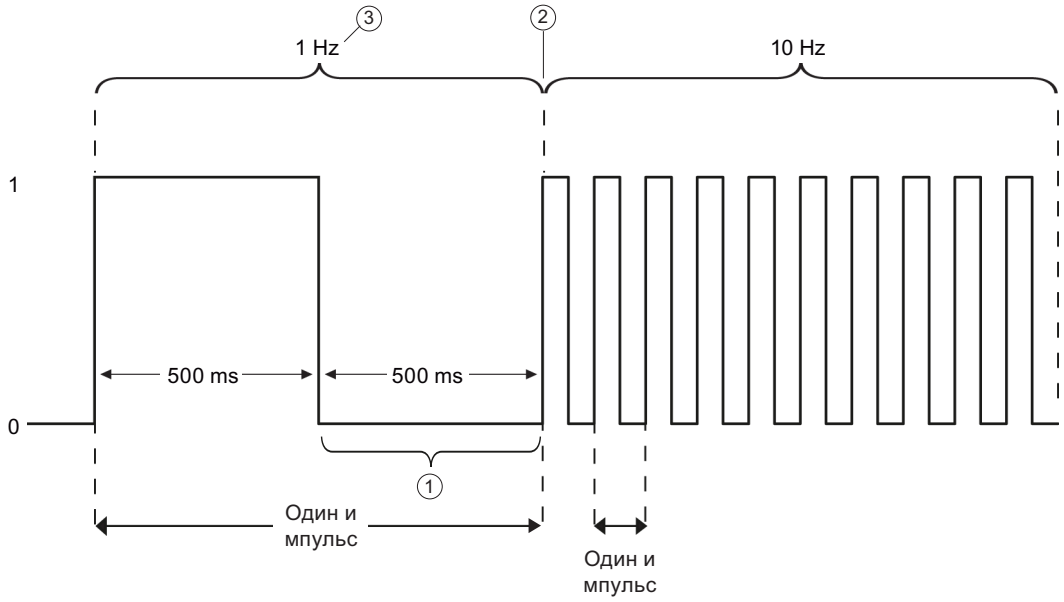
Если установить вход REQ на TRUE, то активируется значение FREQUENCY. Если REQ = FALSE, то выходная частота PTO не может быть изменена и PTO продолжает посылать импульсы.



(1) Без изменений выходной частоты, пока REQ = 0

Так как инструкция CTRL_PTO лишь запускает PTO, инструкция CTRL_PTO сразу же завершается. Поэтому выход BUSY никогда не включается. Выход DONE включается, если нет ошибок. При обнаружении ошибки, параметр ERROR устанавливается на TRUE и параметр STATUS содержит код ошибки.

Если инструкция CTRL_PTO активируется с заданной частотой, то S7-1200 возвращает последовательность импульсов с этой указанной частотой. Требуемая частота может быть изменена в любое время. При изменении частоты, S7-1200 сначала завершает текущий импульс, прежде чем изменить частоту на новое требуемое значение. Пример: Если требуемая частота 1 Гц (это потребует 1000 мс на выполнение) и частота будет через 500 мс изменена на 10 Гц, то изменение частоты будет выполнено в конце промежутка времени в 1000 мс.



① Частота будет изменена через 500 мс на 10 Гц.

② Импульс в 1 Гц должен быть завершен, прежде чем частота сможет быть изменена на новое значение в 10 Гц.

③ 1 Гц соответствует 1000 мс.

Аппаратный объект генератора импульсов имеет следующие ограничения: Только одна инструкция может использовать генератор импульсов как PTO и использование генератора импульсов управляется редактором конфигурации оборудования. Другие инструкции, пытающиеся обращаться к PTO, выводят ошибку: "0x8090" (генератор импульсов с указанным ID оборудования используется).

Примечание

Принудительная активация назначенных PWM и PTO цифровых I/O невозможна.

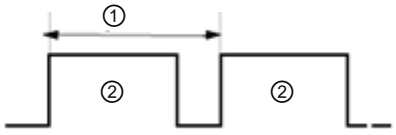
Используемые широтно-импульсной модуляцией (PWM) и выводом последовательности импульсов (PTO) цифровые I/O назначаются при конфигурировании устройства. Если этим функциям назначаются цифровые I/O, то значения адресов назначенных I/O не могут быть изменены с помощью функции принудительного присваивания значений в таблице текущего контроля.

Таблица 9-190 Значения кодов ошибок в параметре STATUS

Код ошибки (W#16# ...)	Описание
0	Ошибки отсутствуют
0x8090	Генератор импульсов с указанным ID оборудования используется.
0x8091	Частота за пределами диапазона. Требуемая частота превышает макс. частоту выбранного импульсного выхода.
0x80A1	Идентификатор PTO (ID оборудования) не адресует действительную PTO.

Код ошибки (W#16# ...)	Описание
0x80D0	Генератор импульсов с указанным ID оборудования не активирован. Активировать генератор импульсов в свойствах CPU в "Генераторы импульсов (PTO/PWM)".
0x80D1	Генератор импульсов с указанным ID оборудования не имеет выбора PTO. Выбрать PTO в конфигурации оборудования.

9.8.3 Принцип работы для импульсных выходов



① Время цикла
② Длительность импульса

Длительность импульса может быть выражена в виде сотых долей времени цикла (от 0 до 100), в виде тысячных долей (от 0 до 1000), в виде десятитысячных долей (от 0 до 10000) или в виде аналогового S7 формата.

Длительность импульса может изменяться от 0 (нет импульса, всегда выключено) до полной шкалы (нет импульса, всегда включено).

Т.к. выход PWM может изменяться от 0 до полной шкалы, то он предлагает цифровой выход, во многом совпадающий с аналоговым выходом. Например, выход PWM может быть использован для управления скоростью двигателя от состояния покоя до максимальной величины, или для управления положением клапана от закрытого состояния до полностью открытого.

Частота настраивается в конфигурации оборудования (Страница 494). Длительность импульса управляется через программу пользователя

Для управления быстрыми импульсными выходами предлагается четыре генератора импульсов: PWM и вывод последовательности импульсов (PTO). PTO используется инструкциями управления перемещением . Каждый генератор импульсов может быть назначен либо PWM, либо PTO, но не обоим одновременно.

Можно использовать встроенные выходы CPU или выходы опциональной сигнальной платы. В таблице ниже перечислены адреса выходов (при этом используется стандартная конфигурация выходов). После изменения адресов выходов, адреса соответствуют вновь назначенным адресам. Следует помнить, что для PWM требуется только один выход, в то время как PTO в качестве опции может использовать два выхода на каждый канал. Если один выход не используется импульсной функцией, то он доступен для других целей. Назначение I/O указано в следующей таблице.

В таблице представлены назначения I/O по умолчанию. Но четыре генератора импульсов могут быть сконфигурированы для любого из встроенных или цифровых выходов SB

модуля CPU. Разные выходы поддерживают разные напряжения и скорости. Это следует учитывать при назначении выходов PWM/PTO.

Примечание

Импульсные последовательности не могут использоваться другими операциями в программе пользователя.

Если выходы CPU или сигнальной платы конфигурируются как генераторы импульсов (для PWM или инструкций управления перемещением PTO), то соответствующие адреса выходов удаляются из памяти выходов и не могут использоваться для других целей в программе пользователя. Если программа пользователя записывает значение на выход, который используется как генератор импульсов, то CPU не записывает это значение на физический выход.

Примечание

Выходы направления PTO могут быть разрешены для использования в другом месте программы.

Для каждого PTO должны быть назначены два выхода: Один выход как импульсный выход и один выход как выход направления. Можно использовать только импульсный выход, но не выход направления. Поэтому можно разрешить выход направления для других целей в программе пользователя.

Таблица 9-191 Назначение выходов по умолчанию для генераторов импульсов[3]

Описание	Импульс	Направление
PTO1		
Встроенные I/O	A0.0	A0.1
SB-I/O	A4.0	A4.1
PWM1		
Встроенные выходы	A0.0	-
Выходы SB	A4.0	-
PTO2		
Встроенные I/O	A0.2	A0.3
SB-I/O	A4.2	A4.3
PWM2		
Встроенные выходы	A0.2	-
Выходы SB	A4.2	-
PTO3		
Встроенные I/O	A0.4[1]	A0.5[1]
SB-I/O	A4.0	A4.1
PWM3		
Встроенные выходы	A0.4[1]	-
Выходы SB	A4.1	-
PTO4		

Описание	Импульс	Направление
Встроенные I/O	A0.6[2]	A0.7[2]
SB-I/O	A4.2	A4.3
PWM4		
Встроенные выходы	A0.6[2]	-
Выходы SB	A4.3	-

[1] У CPU 1211C нет выходов A0.4, A0.5, A0.6 и A0.7. Поэтому нельзя использовать эти выходы для CPU 1211C.

[2] У CPU 1212C нет выходов A0.6 и A0.7. Поэтому нельзя использовать эти выходы для CPU 1212C.

[3] Эта таблица относится к PTO/PWM функциям CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C и CPU 1217C.

9.8.4 Конфигурирование импульсного канала для PWM или PTO

Для подготовки PWM или PTO к работе, сначала следует сконфигурировать импульсный канал в конфигурации устройства путем выбора CPU, генератора импульсов (PTO/PWM) и PWM1/PTO1 до PWM4/PTO4. Активировать генератор импульсов (установить флажок). При активации этому конкретному генератору импульсов присваивается однозначное имя по умолчанию. Это имя может быть изменено в поле редактирования "Имя:", при этом имя должно оставаться однозначным. Имена активированных генераторов импульсов становятся переменными (тегами) в таблице переменных (тегов) "Константы" и могут использоваться как:

- Параметры PWM инструкции CTRL_PWM
- Параметры PTO инструкции CTRL_PTO

В поле редактирования "Комментарий:" можно ввести комментарий для конкретного генератора импульсов.

Таблица 9-192 Выход CPU: Максимальная частота (PTO) и минимальное время цикла (PWM)

CPU	Выходной канал CPU:	Максимальная частота PTO	Минимальное время цикла PWM
1211C	Qa.0 до Qa.3	100 кГц	10 мкс
1212C	Qa.0 до Qa.3	100 кГц	10 мкс
	Qa.4, Qa.5	20 кГц	50 мкс
1214C и 1215C	Qa.0 до Qa.3	100 кГц	10 мкс
	Qa.4 до Qb.1	20 кГц	50 мкс
1217C	DQa.0 до DQa.3 (.0+, .0- до .3+, .3-)	1 МГц	1 мкс
	DQa.4 до DQa.1	100 кГц	10 мкс

Таблица 9-193 Выход сигнальной платы: Максимальная частота (PTO) и минимальное время цикла (PWM)

Сигнальная плата (SB)	Выходной канал SB	Максимальная частота PTO	Минимальное время цикла PWM
SB 1222, 200 кГц	DQe.0 до DQe.3	200 кГц	5 мкс
SB 1223, 200 кГц	DQe.0, DQe.1	200 кГц	5 мкс
SB 1223	DQe.0, DQe.1	20 кГц	50 мкс

Примечание

Минимальное время цикла каждого выхода CPU и сигнальной платы указано в таблицах выше. Но TIA Portal не предупреждает в ситуации, когда конфигурируется генератор импульсов PWM с временем цикла меньше, чем это минимальное время цикла оборудования. Это может вызвать проблемы в приложении. Поэтому всегда следует быть уверенным, что это время цикла не выходит за рамки аппаратных предельных значений.

Примечание

Если устанавливается длительность импульса сигнала PWM, то фактическое время длительности импульса: (время "высокого" состояния и импульса) должно быть больше или равно 1 миллисекунде, т.к. опорным временем являются "Миллисекунды". Если опорным временем являются "Микросекунды", то фактическое время длительности импульса должно быть больше или равно 1 микросекунде. Выход отключается, если длительность импульса меньше 1 "опорного времени".

Пример: Время цикла в 10 микросекунд и длительность импульса в 5 сотых создает время длительности цикла в 0,5 микросекунд. Так как это значение меньше 1 микросекунды, то сигнал PWM выключается.

Параметрирование

В области для параметрирования могут конфигурироваться параметры выходного импульса. Доступны следующие опции, в зависимости от того, выбирается ли PWM или PTO:

- Тип сигнала: Конфигурирование импульсного выхода как PWM или PTO. Более подробная информация о выборе PTO содержится в разделе "Фазировка" (Страница 662).
  - PWM
  - PTO (импульс A и направление B)
  - PTO (импульса прямого счета A и импульс обратного счета B)
  - PTO (A/B со сдвигом по фазе)
  - PTO (A/B со сдвигом по фазе - четыре раза)
- Опорное время (только для PWM): Выбор используемых единиц времени:
  - Миллисекунды
  - Микросекунды

- Формат длительности импульса (только для PWM): Назначение разрешения для длительности импульса (ширины):
  - Сотая часть (от 0 до 100)
  - Тысячная часть (от 0 до 1000)
  - Десятитысячная часть (от 0 до 10000)
  - Аналоговый формат S7 (от 0 до 27648)
- Время цикла (только для PWM): Назначение продолжительности для выполнения цикла (время высокого импульса плюс время низкого импульса соответствует времени цикла). Для изменения времени цикла при работе необходимо установить флажок "Разрешить изменение времени цикла при работе". Более подробная информация содержится в следующем разделе "Адреса I/O". Диапазон от 1 до 16.777.215 единиц времени.
- Длительность начального импульса (только для PWM): Назначение длительности для первого импульса. Для изменения этого значения при работе можно использовать адреса I/O сконфигурированного адреса выходного слова. Диапазон базируется на формате длительности импульса.
- Изменение времени цикла в работе (только для PWM): Если выбрать эту опцию, то программа сможет изменяться время цикла сигнала PWM при работе. Более подробная информация содержится в следующем разделе "Адреса I/O".

Примечание

При установке длительности импульса сигнала PWM необходимо учитывать задержку переключения выходного канала, специфицированную в приложении A. Измеренная на выходе фактическая длительность импульса может превышать выбранную длительность импульса. Увеличение длительности импульса более выражено для импульсов с небольшой длительностью и высокой частотой. Следует проследить, чтобы измеренная на выходе длительность импульса соответствовала бы требованиям пользователя.

Определение значения длительности импульса

"Длительность импульса" получается при умножении "длительности начального импульса" на "время цикла". Если выбирается "опорное время", "формат длительности импульса", "время цикла" и "длительность начального импульса", то следует учитывать, что сумма "длительности импульса" не может быть дробным значением. Если полученная "длительность импульса" является дробным значением, то следует изменить "длительность начального импульса" или "опорное время" для получения целого числа.

Здесь два примера:

- Пример 1: Выбираются следующие значения:
  - Опорное время = Миллисекунды (мс)
  - Формат длительности импульса = Сотая часть (0–100).
  - Время цикла = 3 мс
  - Длительность начального импульса = 75

  Полученная "Длительность импульса" = 0,75 x 3 мс = 2,25 мс
  Это значение "Длительности импульса" является дробным и вызывает ошибку при выполнении инструкции CTRL_PWM. Значение "длительности импульса" должно быть целочисленным.
- Пример 2: Выбираются следующие значения:
  - Опорное времени = Микросекунды (мкс)
  - Формат длительности импульса = Сотая часть (0–100).
  - Время цикла = 3000 мкс
  - Длительность начального импульса = 75

  Полученная "Длительность импульса" = 0,75 x 3000 мкс = 2250 мкс
  Это значение "Длительности импульса" является целым и с ним инструкция CTRL_PWM работает безупречно.

Аппаратные выходы

Выбрать в области аппаратных выходов выходной канал в выпадающем меню. В зависимости от конфигурации, доступен один или два выхода. После назначения выходного канала генератору импульсов, выходной канал больше не может использоваться другим генератором импульсов, HSC или образом процесса.

Примечание

Выходы генератора импульсов не могут использоваться другими инструкциями в программе пользователя.

Если выходы CPU или сигнальной платы конфигурируются как генераторы импульсов (для PWM, PTO или инструкций управления перемещением), то соответствующие выходные адреса удаляются из памяти выходов и не могут использоваться для других целей в программе. Если программа записывает значение на выход, который используется как генератор импульсов, то CPU не записывает это значение на физический выход.

Адреса ввода-вывода

У PWM есть два байта в области памяти A для "Длительности импульса". При работе PWM можно отредактировать значение в назначенной области памяти A, изменив тем самым длительность импульса.

Ввести в области адресов ввода-вывода адрес выходного слова, где должно храниться значение для длительности импульса.

Адресами по умолчанию для значений длительности импульса PWM являются:

- PWM1: AW1000
- PWM2: AW1002
- PWM3: AW1004
- PWM4: AW1006

Для PWM значение по этому адресу управляет длительностью импульса и каждый раз при переходу CPU из STOP в RUN инициализируется с (назначенным выше) значением "Длительность начального импульса:". Значение этого выходного слова изменяется при работе для изменения длительности импульса. Диапазон этого значения зависит от сконфигурированного при параметрировании формата длительности импульса.

Оставшиеся четыре байта в области памяти A также могут быть назначены "Времени цикла" сигнала PWM. См. "Принцип работы для импульсных выходов" (Страница 492) со схемой сигнала PWM. Если флажок "Разрешить изменение времени цикла при работе" установлен, то первые два байта сохраняют значение длительности импульса, а последние четыре байта - значение времени цикла.

При работе PWM можно изменить значение двойного слова на конце назначенной этой PWM области памяти A. При этом изменяется время цикла сигнала PWM. Эта опция активируется, например, чтобы CPU назначил шесть байт для PWM1, и для этого выбирается AB1008 до AB1013. Сразу после загрузки программы и запуска PWM, можно изменить длительность импульса через AW1008 и время цикла через AD1010.

При переходе CPU из STOP в RUN, он инициализирует значение времени цикла в области памяти A с назначенным в разделе "Параметрирование" значением для "Времени цикла". Единицы и диапазоны для значения времени цикла в области памяти A соответствуют конфигурации в разделе "Параметрирование".

Если флажок "Разрешить изменение времени цикла при работе" был установлен, то TIA Portal автоматически выбирает новый адрес для адреса вывода. Новый адрес вывода не может совпадать с адресом по умолчанию для этого генератора импульсов. TIA Portal использует следующий доступный блок с шестью следующими друг за другом байтами. Если при поиске в области памяти A доступный блок не будет найден до конца области памяти, то поиск начинается заново с адреса "0" в области памяти A и продолжается до нахождения доступного блока.

Сконфигурированный для PTO генератор импульсов не использует адрес выходного слова.

9.9 Рецептуры и архивы данных

9.9.1 Рецептуры

9.9.1.1 Обзор рецептур

Сохранение данных рецептур

- Блок данных рецептур, создаваемый в проекте, должен быть сохранен в **загружаемой** памяти CPU. Для этого можно использовать внутреннюю память CPU или внешнюю карту памяти, используемую в качестве программной карты памяти.
- Еще один DB, который должен быть создан, это активный блок данных рецептур. Этот DB должен находиться в **рабочей** памяти, где соответствующий активный рецептурный набор данных (запись) может быть считан или записан логикой программы.

Управление данными рецептур

Рецептурный DB использует массив из записей рецептур продуктов. Каждый элемент массива рецептур представляет собой отдельный вариант рецептуры, использующий общий набор компонентов.

- Создается тип данных PLC или тип данных Struct, определяющий все компоненты в рецептурной записи. Этот шаблон типа данных используется повторно для всех записей рецептур. Рецептуры продукта варьируются в зависимости от начальных значений, которые назначены компонентам рецепта.
- Любая из рецептур, при необходимости, с помощью инструкции READ_DBL может быть передана из рецептурного DB (все рецептуры в загружаемой памяти) в активный рецептурный DB (одна рецептура в рабочей памяти). После перемещения записи рецептуры в рабочую память, логика программы может начать считывать значения компонентов и запустить производство. Передача минимизирует объем необходимой для хранения рецептурных данных рабочей памяти CPU.
- Если значения компонента активной рецептуры корректируются устройством HMI во время производственного цикла, то измененные значения могут быт записаны обратно в рецептурный DB с помощью инструкции WRIT_DBL.

Экспорт рецептур (из рецептурного DB в файл CSV)

С помощью инструкции RecipeExport весь набор записей рецептур может быть экспортирован в файл CSV. Неиспользуемые записи рецептур также будут экспортированы.

Импорт рецептур (из файла CSV в рецептурный DB)

После экспорта рецептур созданный файл CSV может использоваться как шаблон структуры данных.

1. Загрузить имеющийся рецептурный файл CSV через страничку браузера файлов веб-сервера CPU из CPU на PC.
2. С помощью текстового ASCII-редактора отредактировать рецептурный файл CSV. Можно изменяться стартовые значения компонентов, но не типы или структуру данных.
3. Выгрузить измененный файл CSV из PC обратно в CPU. Но для этого придется удалить старый файл CSV (с тем же именем) из загружаемой памяти CPU или переименовать его, чтобы веб-браузер CPU мог выполнить выгрузку.
4. После выгрузки измененного файла CSV в CPU, новые стартовые значения с помощью инструкции RecipeImport могут быть переданы из измененного файла CSV (в загружаемой памяти CPU) в рецептурный DB (в загружаемой памяти CPU).

9.9.1.2 Пример для рецептуры

Пример для рецептур

В таблице представлен процесс подготовки рецептурной информации для использования в рецептурном DB. В этом примере для рецептурного DB сохранено пять записей (наборов данных), из которых используются три. Четвертый и пятый набор данных зарезервированы для последующих расширений. Каждая строка в таблице представляет собой набор данных (запись) с названием рецептуры, типам данных компонентов и значениями компонентов.

productname	water	barley	wheat	hops	yeast	waterTmp	mashTmp	mashTime	QTest
Pils	10	9	3	280	39	40	30	100	0
Lager	10	9	3	150	33	50	30	120	0
BlackBeer	10	9	3	410	47	60	30	90	1
Not_used	0	0	0	0	0	0	0	0	0
Not_used	0	0	0	0	0	0	0	0	0

Создание рецептурного блока данных

Примечание

Правила для рецептурных блоков данных

- Рецептурный DB должен содержать одномерный массив данных типа PLC или типа Struct. В примере рецептуры показано создание рецептурного DB с типом данных PLC.
- В примере все ингредиенты компонентов имеют тип данных UINT. Типы данных компонентов могут представлять собой и комбинацию различных типов данных, за исключением Struct. Struct не разрешен в элементе массива в рецептурном DB ни для типа данных PLC, ни для типа данных Struct.

Первый шаг: Создание нового типа данных PLC

Добавить новый тип данных PLC с названием, соответствующим типу рецептуры. На рисунке ниже "Beer_Recipe" - это новый сложный тип данных PLC, включающий в себя последовательность простых типов данных. Тип данных PLC "Beer_Recipe" это шаблон данных, который используется к каждой записи рецептурного DB и в активном рецептурном DB. Ввести названия компонентов и типы данных, общие для всех примеров рецептур. Отдельные значения компонентов после будут добавлены в рецептурный DB.

Beer_Recipe

	Name	Data type	Default value
1	productname	String[20]	'Beer_Recipe'
2	water	UInt	0
3	barley	UInt	0
4	wheat	UInt	0
5	hops	UInt	0
6	yeast	UInt	0
7	waterTmp	UInt	0
8	mashTmp	UInt	0
9	mashTime	UInt	0
10	QTest	UInt	0

Второй шаг: Создание рецептурного блока данных

- Создать рецептурный DB как глобальный блок данных с активированным свойством DB "Сохранение только в загружаемой памяти".
- Название рецептурного блока данных будет использоваться в качестве имени соответствующего файла CSV. Содержащиеся в названии DB символы должны отвечать правилам для присвоения имен в файловой системе Windows. Символы \ / : * ? " < > | и пробел запрещены.
- Назначение массива рецептур - это "Products" как Array [1.. 5] of "Beer_Recipe". Размер массива 5 это максимально возможное число вариантов рецептур.
- Значения для компонентов рецептур добавляются как стартовые значения DB.

На следующем рисунке приведена расширенная рецептура "BlackBeer", чтобы показать все компоненты записи рецептуры.

Recipe_DB

	Name	Data type	Offset	Start value
1	Static			
2	Products	Array [1 .. 5] of "Beer_Recipe"	...	
3	Products[1]	"Beer_Recipe"	...	
4	Products[2]	"Beer_Recipe"	...	
5	Products[3]	"Beer_Recipe"	...	
6	productname	String[20]	...	'BlackBeer'
7	water	UInt	...	10
8	barley	UInt	...	9
9	wheat	UInt	...	3
10	hops	UInt	...	410
11	yeast	UInt	...	47
12	waterTmp	UInt	...	60
13	mashTmp	UInt	...	30
14	mashTime	UInt	...	90
15	QTest	UInt	...	1
16	Products[4]	"Beer_Recipe"	...	
17	Products[5]	"Beer_Recipe"	...	

Экспорт рецептур (из рецептурного DB в файл CSV)

При выполнении инструкции "RecipeExport (Страница 504)" данные из рецептурного DB передаются в файл CSV (см. следующий текстовый файл).

```
Recipe_DB.csv
index,productname,water,barley,wheat,hops,yeast,waterTmp,
mashTmp,mashTime,QTest
1,”Pils”,10,9,3,280,39,40,30,100,0
2,”Lager”,10,9,3,150,33,50,30,120,0
3,”BlackBeer”,10,9,3,410,47,60,30,90,1
4 ”Not_used”,0,0,0,0,0,0,0,0,0
5 ”Not_used”,0,0,0,0,0,0,0,0,0
```

Импорт рецептур (из файла CSV в рецептурный DB)

1. Имеющийся рецептурный файл CSV загружается через страничку браузера файлов (Страница 1068) веб-сервера CPU из CPU на PC.
2. С помощью текстового ASCII-редактора отредактировать рецептурный файл CSV. Можно изменяться стартовые значения компонентов, но не типы или структуру данных.
3. Выгрузить измененный файл CSV из PC обратно в CPU. Но для этого придется удалить старый файл CSV (с тем же именем) из загружаемой памяти CPU или переименовать его, чтобы веб-браузер CPU мог выполнить выгрузку.
4. После выгрузки измененного файла CSV в CPU, новые стартовые значения с помощью инструкции RecipeImport могут быть переданы из измененного файла CSV (в загружаемой памяти CPU) в рецептурный DB (в загружаемой памяти CPU).

Файлы CSV по структуре должны полностью совпадать с соответствующим рецептурным DB

- Можно изменять значения в файле CSV, но структура должна оставаться неизменной. Обязательным условиям для инструкции RecipeImport является точное соответствие числа записей и компонентов структуре целевого рецептурного DB. Иначе инструкция RecipeImport не будет выполнена. Пример: В рецептурном DB определено 10 рецептур, но используются из них только 6, несмотря на это, строки с 7 по 10 файла CSV также будут переданы в DB. Необходимо определить, являются ли эти данные действительными или нет. Так, например, в неиспользуемых рецептурных записях названию продукта может быть присвоена переменная “Not_used”.
- При добавлении записей данных в текстовый файл и импорте измененного файла следует помнить, что назначенное предельное значение массива рецептурного DB должно содержать достаточное число элементов для всех рецептурных записей.
- При экспорте в файл CSV номер индекса создается автоматически. Если создаются дополнительные записи данных, то следует добавлять соответствующие номера индекса.
- При выполнении инструкции RecipeImport файл CSV проверяется, имеет ли он правильную структуру и соответствуют ли значения типам данных, заданным в соответствующем рецептурном DB. Например, тип данных Bool не может хранить целочисленное значение, и выполнение инструкция RecipeImport в таком случае завершается с ошибкой.

Отображение рецептурных данных файла CSV в Excel

Для удобства чтения и редактирования файл CSV может быть открыт в Excel. Если запятые не определяются как десятичные разделители, то следует использовать функцию импорта программы Excel для структурированного вывода данных.

	A	B	C	D	E	F	G	H	I	J	K
1	index	product	water	barley	wheat	hops	yeast	waterTmp	mashTmp	mashTime	QTest
2	1	"Pils"	10	9	3	280	39	40	30	100	0
3	2	"Lager"	10	9	3	150	33	50	30	120	0
4	3	"BlackBeer"	10	9	3	410	47	60	30	90	1
5	4	"Not_used"	0	0	0	0	0	0	0	0	0
6	5	"Not_used"	0	0	0	0	0	0	0	0	0

Примечание

Запятые в поле имени элемента типа данных PLC

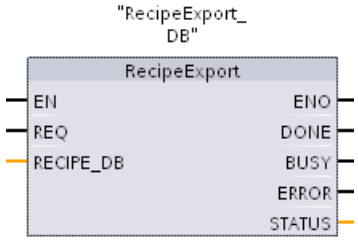
Не вводить в поле имени используемого в рецептуре элемента типа данных PLC запятых. При вводе запятых в поле имени, программа Excel добавляет столбцы в отображаемый файл CSV. Эти столбцы могут стать причиной ошибок при редактировании стартовых значений файла рецептурной записи.

9.9.1.3 Программные инструкция для передачи рецептурных данных

RecipeExport (экспорт рецептуры)

Таблица 9-194 Инструкция RecipeExport

LAD/FBD	SCL	Описание

	`"RecipeExport_DB"(`
	`req:=_bool_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	
	`Recipe_DB:=_variant_inout_);`
		Инструкция "RecipeExport" экспортирует рецептурные записи из рецептурного блока данных в формат файла CSV. Файл CSV содержит название продукта, имена компонентов и стартовые значения. Файл CSV сохраняется во внутреннюю загружаемую память или во внешнюю загружаемую память, если установлена опциональная внешняя карта памяти, используемая как программная карта.
		Процесс экспорта запускается параметром "REQ". Параметр BUSY при обработке экспортной операции устанавливается на 1. После выполнения RecipeExport, параметр BUSY сбрасывается на 0, и завершение процесса отображается на параметре DONE с 1. При возникновении ошибки при выполнении, параметры ERROR и STATUS показывают результат.

Условие для выполнения экспорта рецептур является создание рецептурного DB. Название рецептурного блока данных будет использоваться в качестве имени нового файла CSV. Если уже есть файл CSV с таким именем, то он заменяется в процессе экспорта.

Через страничку браузера файлов (Страница 1068) встроенного веб-сервера CPU можно обращаться к файлу CSV с рецептурами. Файл сохраняется в папке рецептур в корневой директории загружаемой памяти CPU.

Таблица 9-195 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Управляющий параметр REQUEST: Активирует экспорт при положительном фронте.
RECIPE_DB	In/Out	Variant	Указатель на рецептурный блок данных. Более подробная информация содержится в разделе Пример для рецептурного DB (Страница 500). Содержащиеся в названии DB символы должны отвечать правилам для присвоения имен в файловой системе Windows. Символы \ / : * ? " < > | и пробел запрещены.
DONE	OUT	Bool	Бит DONE = TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок. (значение по умолчанию: False)

Параметр и тип данных		Тип данных	Описание
BUSY	OUT	Bool	Выполнение RecipeExport • 0: Нет обрабатываемых инструкций • 1: Инструкция обрабатывается
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение TRUE, после того, как последний запрос был завершен с ошибкой. Код ошибки в параметре STATUS действителен только в цикле, где ERROR = TRUE. • 0: Нет предупреждений и ошибок • 1: Произошла ошибка. Параметр STATUS содержит информацию о типе ошибки.
STATUS	OUT	Word	Условие выполнения

Таблица 9-196 Значения ERROR иSTATUS

ERROR	STATUS (W#16#....)	Описание
0	0000	Ошибки отсутствуют
0	7000	Вызов без REQ фронта: BUSY = 0, DONE = 0
0	7001	Первый вызов с REQ фронтом (в обработке): BUSY = 1, DONE = 0
0	7002	N-й вызов (в обработке): BUSY = 1, DONE = 0
1	8070	Вся память экземпляра занята.
1	8090	В имени файла содержатся недопустимые символы.
1	8091	Невозможно обработать указанную в RECIPE_DB структуру данных.
1	8092	Указанная в RECIPE_DB структура данных превышает 5000 байт.
1	80B3	Недостаточно места на MC или во внутренней загружаемой памяти.
1	80B4	Карта памяти имеет защиту от записи.
1	80B6	Атрибут "Сохранять только в загружаемой памяти" рецептурного DB не активирован.
1	80C0	Файл CSV временно заблокирован.
1	80C1	DB временно заблокирован.

RecipeImport (импорт рецептур)

Таблица 9-197 Инструкция RecipeImport

LAD/FBD	SCL	Описание

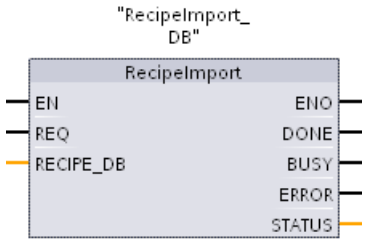
	`"RecipeImport_DB"(`
	`req:=_bool_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	
	`Recipe_DB:=_variant_inout_);`
		Инструкция "RecipeImport" импортирует рецептурные данные из файла CSV в загружаемой памяти CPU в указанный параметром RECIPE_DB рецептурный блок данных. В процессе импорта стартовые значения в рецептурном блоке данных переписываются. Процесс импорта запускается параметром "REQ". Параметр BUSY при обработке импортной операции устанавливается на 1. После выполнения RecipeImport, параметр BUSY сбрасывается на 0, и завершение процесса отображается на параметре DONE с 1. При возникновении ошибки при выполнении, параметры ERROR и STATUS показывают результат.

Таблица 9-198 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Управляющий параметр REQUEST: Активирует импорт при положительном фронте.
RECIPE_DB	In/Out	Variant	Указатель на рецептурный блок данных. Более подробная информация содержится в разделе "Пример для рецептурного DB (Страница 500)". Содержащиеся в названии DB символы должны отвечать правилам для присвоения имен в файловой системе Windows. Символы \ / : * ? " < > | и пробел запрещены.
DONE	OUT	Bool	Бит DONE = TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок. (значение по умолчанию: False)
BUSY	OUT	Bool	• 0 - Нет обрабатываемых инструкций • 1 - Инструкция обрабатывается
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение TRUE, после того, как последний запрос был завершен с ошибкой. Код ошибки в параметре STATUS действителен только в цикле, где ERROR = TRUE.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

Для выполнение процесса импорта необходимо наличие рецептурного DB со структурой, соответствующей структуре данных файла CSV.

Правила для файлов CSV:

- Файл CSV должен находиться в корневой директории "Recipes" во внутренней загружаемой памяти или во внешней загружаемой памяти, если установлена опциональная внешняя карта памяти, используемая как программная карта.
- Имя файла CSV должно совпадать с именем блока данных в параметре RECIPE_DB.
- Первая строка (заголовок) файла CSV содержит название компонентов рецептуры. Первая строка при импорте игнорируется. Названия компонентов рецептуры в файле CSV и в блоке данных в процессе импорта не согласуются друг с другом.
- В любом случае первое значение в каждой строке файла CSV является номером индекса рецептуры. Отдельные рецептуры импортируются в последовательности индекса. Для этого индекс в файле CSV должны возрастать и не содержать пробелов (в этом случае в параметре STATUS выдается сообщение об ошибке 80B0).
- В файле CSV не должно содержаться больше рецептурных записей, чем предусмотрено в рецептурном блоке данных. Максимальное число записей указывается предельными значениями массива в блоке данных.

Таблица 9-199 Значения ERROR иSTATUS

ERROR	STATUS (W#16#....)	Описание
0	0000	Ошибки отсутствуют
0	7000	Вызов без REQ фронта: BUSY = 0, DONE = 0
0	7001	Первый вызов с REQ фронтом (в обработке): BUSY = 1, DONE = 0
0	7002	N-й вызов (в обработке): BUSY = 1, DONE = 0
1	8070	Вся память экземпляра занята.
1	8090	В имени файла содержатся недопустимые символы.
1	8092	Подходящий файл CSV для импорта не найден. Возможная причина: Имя файла CSV не совпадает с именем рецептурного блока данных.
1	80C0	Файл CSV временно заблокирован.
1	80C1	Блок данных временно заблокирован.
1	80B0	Нумерация в индексе файла CSV является не сквозной, не увеличивающейся или превышает максимальное количество (границу массива) в блоке данных.
1	80B1	Структуры рецептурного блока данных и файла CSV не совпадают: Файл CSV содержит слишком много полей.
1	80B2	Структуры рецептурного блока данных и файла CSV не совпадают: Файл CSV содержит слишком мало полей.
1	80B6	Атрибут "Сохранять только в загружаемой памяти" рецептурного DB не активирован.
1	80D0 +n	Структуры рецептурного блока данных и файла CSV не совпадают: Тип данных в поле "n" не совпадает (n<=46).
1	80FF	Структуры рецептурного блока данных и файла CSV не совпадают: Тип данных в поле "n" не совпадает (n > 46).

9.9.1.4 Пример программы для работы с рецептурами

Необходимые условия для примера программы работы с рецептурами

Для примера программы работы с рецептурами должны быть выполнены следующие условия:

- Рецептурный DB со всеми рецептурными записями. Рецептурный DB находится в загружаемой памяти.
- Активный рецептурный DB с копией одной рецептуры в рабочей памяти.

Более подробная информация о рецептурном DB и соответствующем файле CSV содержится в разделе "Пример для рецептурного DB (Страница 500)".

Создание активного рецептурного DB

В окне "Добавление нового блока":

- Нажать на экранную кнопку "Блок данных" в окне "Добавление нового блока".
- Выбрать в выпадающем списке "Тип" прежде созданный тип данных PLC "Beer_recipe".

Стартовые значения не требуются. При передаче рецептуры из рецептурного DB в активный рецептурный DB, устанавливаются значения данных DB. В примере активный рецептурный DB является целью для данных инструкции READ_DBL и предлагает исходные данные для инструкции WRITE_DBL. На изображении ниже показан DB "Active_Recipe".

Active_Recipe

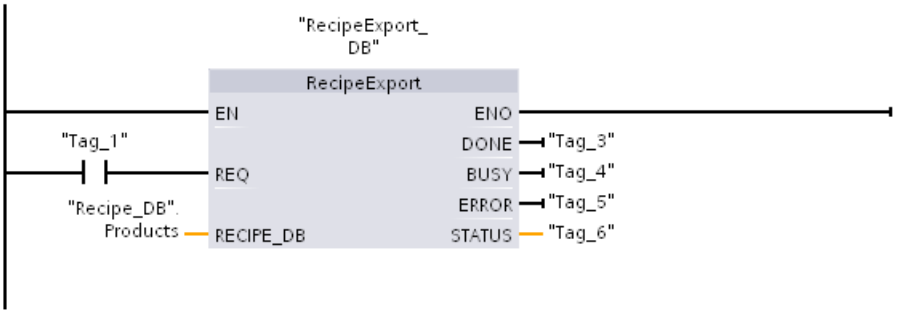
	Name	Data type	Start value
1	Static		
2	productname	String[20]	'Beer_Recipe'
3	water	UInt	0
4	barley	UInt	0
5	wheat	UInt	0
6	hops	UInt	0
7	yeast	UInt	0
8	waterTmp	UInt	0
9	mashTmp	UInt	0
10	mashTime	UInt	0
11	QTest	UInt	0

DB экземпляра

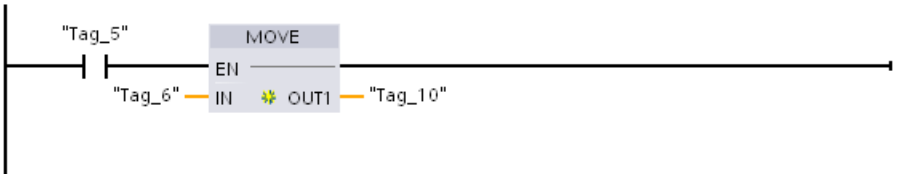
Используемые инструкциями RecipeExport ("RecipeExport_DB") и RecipeImport ("RecipeImport_DB") DB экземпляра создаются автоматический при вставке инструкций в программу. DB экземпляра используются для управления выполнением инструкций и не упоминаются в логике программы.

Пример программы для работы с рецептурами

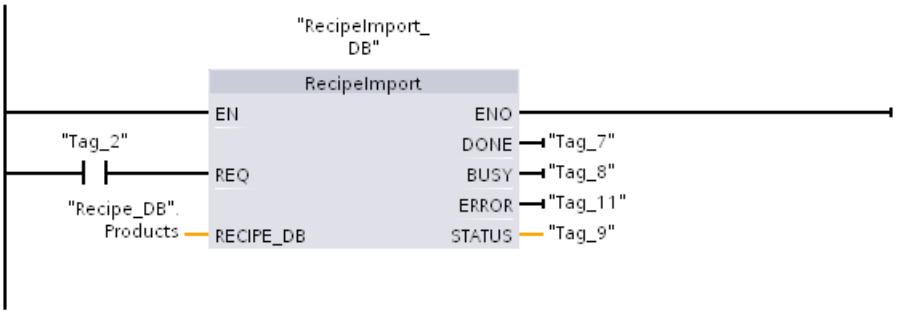
Сегмент 1 Положительный фронт на REQ запускает процесс экспорта. Из данных рецептурного DB создается файл CSV и помещается в папку рецептур в памяти CPU.



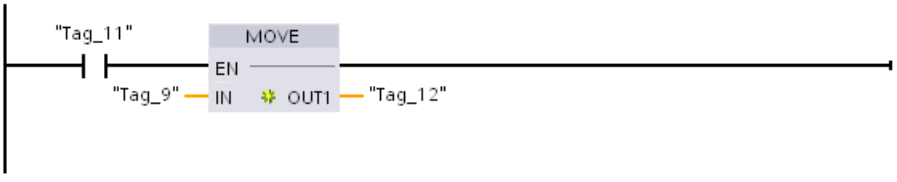
Сегмент 2 Снять выход STATUS из выполнения RecipeExport, потому что он действителен только в течение одного цикла.



Сегмент 3 Положительный фронт на REQ запускает процесс импорта. В имеющийся рецептурный DB загружаются все считанные из соответствующего файла CSV, находящегося в папке рецептур памяти CPU, рецептурные данные.

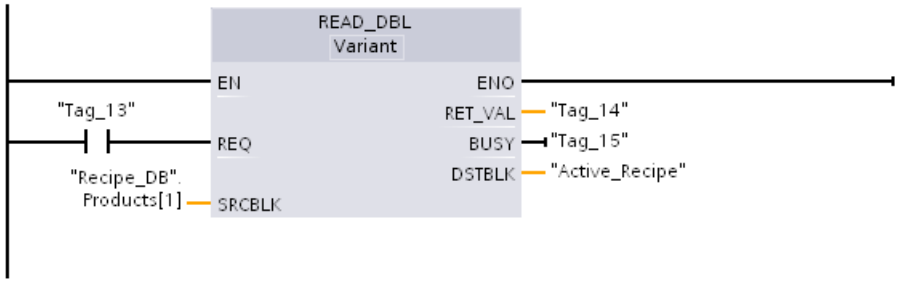


Сегмент 4 Снять выход STATUS из выполнения RecipeImport, потому что он действителен только в течение одного цикла.

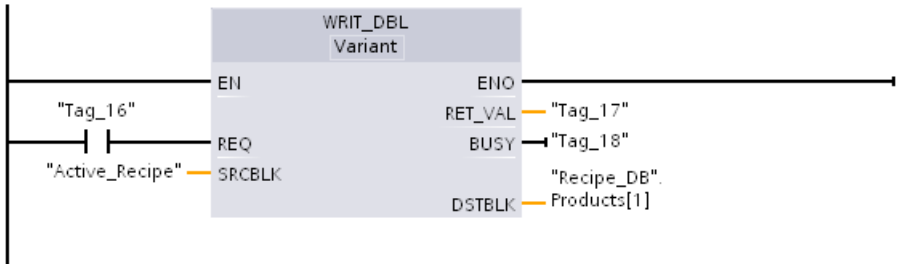


Сеть 5 READ_DBL копирует стартовые значения из рецептуры "Recipe_DB". Products[1] (в загружаемой памяти CPU) в текущие значения DB "Active_Recipe" (в рабочей памяти CPU). После выполнения READ_DBL логика программы через адреса в памяти DB "Active_Recipe" может обращаться к значениям компонентов рецептуры. Так, к примеру, символьные

адреса ("Active_Recipe".productname) и ("Active_Recipe".water) логики программы указывают актуальное имя рецептуры и количество воды.



Сегмент 6 Во время рабочего цикла, с помощью HMI-устройства можно изменить значение компонента, сохраненного в DB "Active_Recipe". Измененные рецептурные данные можно сохранить, выполнив инструкцию WRIT_DBL. В примере все стартовые значения Recipe_DB для одной рецептуры "Recipe_DB". Products[1] перезаписываются фактическими значениями из DB "Active_Recipe".



9.9.2 Архивы данных

Управляющая программа может использовать инструкции Data log для сохранения значений данных рабочего цикла в постоянных архивных файлах. CPU сохраняет файла архивов данных во флэш-памяти (в CPU или на карте памяти) в стандартном формате CSV (значения, разделенные запятой). Записи данных организованы в виде циклически обновляемого архивного файла предварительно заданного размера.

С помощью инструкций Data log в программе можно создавать, открывать, закрывать архивные файлы, а также записывать наборы данных в файлы. Для выбора архивируемых значений программы создается буфер данных, определяющий один единственный архивный набор данных. CPU использует свой буфер данных в качестве временной памяти для нового архивного набора данных. Управляющая программа в рабочем цикле перемещает новые актуальные значения в буфер. После обновления программой всех актуальных значений данных, она может выполнить инструкцию DataLogWrite для переноса данных из буфера в архивный набор данных.

Файлы архива данных могут открываться, редактироваться, сохраняться, переименовываться и удаляться на страничке браузера файлов веб-сервера. Для отображения браузера файлов необходим доступ по чтению, а для редактирования, удаления или переименования файлов архива данных - доступ по записи.

9.9.2.1 Структура набора данных архива

Параметры DATA и HEADER инструкции DataLogCreate задают тип данных и описание заголовков столбцов для всех элементов данных в архивном наборе данных.

Параметр DATA для инструкции DataLogCreate

Параметр DATA адресует память, используемую в качестве временного буфера для нового архивного блока данных. Ему должен быть назначен M- или DB-адрес.

Может быть назначен как полный DB (производный от типа данных PLC, указанного при создании DB), так и часть DB (в случае указанного элемента DB речь может идти о любом типе данных, структуре типа данных, типе данных PLC или массиве данных).

Для инструкции DataLogCreate структурные типы данных имеют ограничение до одного единственного уровня вложенности. Массив из символьных строк в этом контексте не считается отдельным уровнем вложенности. В настоящее время инструкция DataLogCreate не возвращает ошибку. Инструкция обрабатывает только первую строку символов массива. Общее число объявленных элементов данных должно соответствовать числу указанных в параметре HEADER столбцов. Может быть назначено максимум 253 элемента данных (с меткой времени) или 255 элементов данных (без метки времени). Благодаря такому ограничению набор данных не выходит за границы значения в максимум 256 столбцов для листа Excel.

Параметр DATA может назначать сохраняемые или несохраняемые элементы данных в стандартном ("Standard") (совместимом с S7-300/400) или оптимизированном типе DB.

Для записи архивного набора данных, сначала необходимо загрузить новые технологические значения во временный набор данных DATA и после выполнить инструкцию DataLogWrite, после чего значения набора данных будут сохранены в файл архива данных.

Параметр HEADER для инструкции DataLogCreate

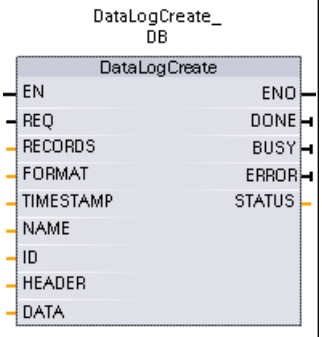
Параметр HEADER указывает на заголовки столбцов в верней строке матрицы данных в файле CSV. HEADER данные должны находиться в DB или M памяти и символы должны отвечать действующим правилам форматирования для файлов CSV, когда отдельные имена столбцов разделяются запятыми. В случае типа данных речь может идти о строках, байтовых массивах или символьных массивах. Символьные и байтовые массивы позволяют увеличить размер, в отличие от строкового типа данных, ограниченного максимум 255 байтами. Параметр HEADER является необязательным. Если HEADER не назначен, то строка заголовка не создается в файле архива данных.

9.9.2.2 Программные инструкции для управления архивами данных

DataLogCreate (создание архива данных)

Таблица 9-200 Инструкция DataLogCreate

LAD/FBD	SCL	Описание

	`"DataLogCreate_DB"(`
	`req:=_bool_in_,`
	`records:=_udint_in_,`
	`format:=_uint_in_,`
	`timestamp:=_uint_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`name:=_string_inout_,`
	`ID:=_dword_inout_,`
	`header:=_variant_inout_,`
	`data:=_variant_inout_);`
		Создание и инициализация файла архива данных. CPU создает файл в папке \DataLogs, использует имя в параметре NAME и не явно открывает файл для процессов записи. С помощью инструкций Data log технологические данные рабочего цикла могут сохраняться в программе во флэш-память CPU или на карту памяти.
		При добавлении инструкции, STEP 7 автоматически создает соответствующий DB экземпляра.

[1] В примере SCL "DataLogCreate_DB" - это имя DB экземпляра.

Таблица 9-201 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Инструкцию запускает положительный фронт (0 на 1). (значение по умолчанию: False)
RECORDS	IN	UDint	Максимальное количество наборов данных, которое может содержать круговой архив данных перед заменой самой старой записи: Набор данных с заголовком не входит в это число. Для успешного создания архива данных необходим достаточный объем загружаемой памяти PLC. (предустановка - 1)
FORMAT	IN	UInt	Формат архива данных: • 0 - внутренний формат (не поддерживается) • 1 - значения, разделенные запятой, "csv-eng" (значение по умолчанию)
TIMESTAMP	IN	UInt	Формат отметки времени данных: Заголовок столбца для полей даты и времени указывается по желанию. Метка времени может ссылать на системное время (Coordinated Universal Time - UTC) или местное время. • 0 - нет метки времени • 1 - метка даты и времени, системное время (по умолчанию) • 2 - метка даты и времени, местное время

Параметр и тип данных		Тип данных	Описание
NAME	IN	Variant	Имя архива данных: Вводится имя. Этот вариант поддерживает только тип данных String и может быть сохранен только в DB или локальной памяти. (значение по умолчанию: ' ') Эта строка символов является именем файла архива данных. Следует использовать ASCII символы, символы \ / : * ? " < > | и пробел запрещены.
ID	In/Out	DWord	Цифровой идентификатор архива данных: Это сгенерированное значение сохраняется для использования с другими инструкциями архива данных. Параметр ID используется для инструкции DataLogCreate только как выход. (значение по умолчанию: 0) Доступ к этому параметру по символьному имени недопустим.
HEADER	In/Out	Variant	Указатель на заголовки столбцов архива данных в верней строке матрицы данных в файле CSV. (значение по умолчанию: ноль). HEADER данные должны находиться в DB или M памяти. Символы должны отвечать действующим правилам форматирования для файлов CSV, и отдельные имена столбцов разделяются запятыми. В случае типа данных речь может идти о строках, байтовых массивах или символьных массивах. Символьные и байтовые массивы позволяют увеличить размер, в отличие от строкового типа данных, ограниченного максимум 255 байтами. Параметр HEADER является необязательным. Если HEADER не задан, то строка заголовка не создается в файле архива данных.
DATA	In/Out	Variant	Указатель на структуру данных набора данных, пользовательский тип данных (UDT) или массив. Данные наборов данных должны находиться в DB или M памяти. Параметр DATA указывает отдельные элементы данных (столбцы) архивного набора и их тип данных. Для инструкции DataLogCreate структурные типы данных имеют ограничение до одного единственного уровня вложенности. Массив из символьных строк в этом контексте не считается отдельным уровнем вложенности. В настоящее время инструкция DataLogCreate не возвращает ошибку. Инструкция обрабатывает только первую строку символов массива. Число объявленных элементов данных должно соответствовать числу указанных в параметре HEADER столбцов. Может быть назначено максимум 253 элемента данных (с меткой времени) или 255 элементов данных (без метки времени). Благодаря такому ограничению набор данных не выходит за границы значения в максимум 256 столбцов для листа Excel.
DONE	OUT	Bool	Бит DONE = TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок. (значение по умолчанию: False)

Параметр и тип данных		Тип данных	Описание
BUSY	OUT	Bool	• 0 - Нет обрабатываемых инструкций • 1 - Инструкция обрабатывается
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение TRUE, после того, как последний запрос был завершен с ошибкой. Код ошибки в параметре STATUS действителен только в цикле, где ERROR = TRUE.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

CPU на основе параметров RECORDS и DATA создает файл архива данных с заданным фиксированным размером и кольцевой структурой наборов данных. Инструкция DataLogCreate назначает постоянную память CPU для всего архива данных, если инструкция DONE = TRUE. Необходимая область памяти в CPU превышает размер файла из-за ресурсов управления файловой системой и сопряженных значений. Постоянная память для архива данных остается выделенной до тех пор, пока CPU снова не освободи ее одним из следующих способов:

- Программа пользователя вызывает инструкцию DataLogDelete.
- Пользователь веб-сервера удаляет архив данных на веб-сервере.
- Пользователь SIMATIC Automation Tool удаляет архив данных в SIMATIC Automation Tool.

Если файл архива данных удаляется другим способом, например, с помощью картридера, то постоянная память в архива данных в CPU не освобождается.

Инструкция DataLogWrite вставляет новые наборы данных в файл архива данных до тех пор, пока не будет сохранено максимальное число блоков данных, заданное в параметре RECORDS. После следующий набор данных перезаписывает самый старый набор данных. Следующая инструкция DataLogWrite перезаписывает следующий за самым старым набор данных и т. д.

Использование ресурсов памяти:

- Архивы данных занимают только загружаемую память.
- Совокупный размер всех архивов данных ограничен доступной емкостью загружаемой памяти. Одновременно может быть открыто восемь архивов. Для управления своими архивами данных можно использовать браузер файлов Браузер файлов (Страница 1068) на стандартной веб-странице. Информацию о том, сколько архивов данных может сохраняться одновременно, можно найти в описании этой стандартной веб-страницы.
- Макс. возможное число для параметра RECORDS это предельное значение числа UDint (4.294.967.295). Фактическое предельное значение для параметра RECORD зависит от размера отдельного набора данных, размера других архивов данных и доступной емкости загружаемой памяти. Кроме этого, Excel ограничивает число строк на листе.

Примечание

Архив данных должен быть создан до момента запуска инструкции для записи архива данных

- Операции DataLogCreate и DataLogNewFile для создания архивов данных растягиваются на несколько программных циклов. Фактически необходимое для создания файла архива время зависит от структуры набора данных и количества наборов данных. Логика программы должна отслеживать и регистрировать переключение бита DONE на TRUE, что является сигналом завершения создания для файла архива. Если программа пользователя выполнит инструкцию DataLogWrite до завершения создания архива данных, то операция записи нового архивного набора данных может пройти не так, как положено.
- В определенных ситуациях, когда цикл выполняется очень быстро, создание архива данных может занять больше времени. Если формирование архива данных происходит слишком медленно, то следует убедиться, что флажок для активации минимального времени цикла для циклически выполняемых OB установлен и минимальное время цикла равно 1 мс или выше. Дополнительную информацию можно найти в Конфигурирование времени цикла и коммуникационной нагрузки (Страница 107).

Примечание

Инструкция DataLogNewFile копирует структуру набора данных существующего архива данных

Если перезапись наборов данных является нежелательной, то с помощью инструкции DataLogNewFile можно создать новый архив данных на основе текущего архива данных, после того, как в текущий архив будет записано максимальное число наборов данных. После этого новые наборы данных будут записываться в новый файл архива данных. Старый файл архива данных и данные блоков данных останутся во флэш-памяти.

Таблица 9-202 Значения ERROR иSTATUS

ERROR	STATUS (W#16#....)	Описание
0	0000	Ошибки отсутствуют
0	7000	Вызов без REQ фронта: BUSY = 0, DONE = 0
0	7001	Первый вызов с REQ фронтом (в обработке): BUSY = 1, DONE = 0
0	7002	N-й вызов (в обработке): BUSY = 1, DONE = 0
1	8070	Общая внутренняя экземплярная память занята.
1	807F	Внутренняя ошибка
1	8090	Недействительное имя файла
1	8091	Параметр имени не является строкой.
1	8093	Архив данных с таким именем уже существует. Использовать другое имя, убедиться, что файл CSV архива данных не открыт и после удалить существующий архив данных через страничку браузера файлов Браузер файлов (Страница 1068) веб-сервера.

ERROR	STATUS (W#16#....)	Описание
1	8097	Требуемая длина файла превышает макс. размер файла для файловой системы.
1	80B2	Отсутствуют дополнительные ID ресурсов Примечание: Для исключения этой ошибки следует удалить несколько имеющихся архивов данных или уменьшить число столбцов в структуре наборов данных.
1	80B3	Недостаточно свободной загружаемой памяти.
1	80B4	Карта памяти (MC) имеет защиту от записи.
1	80C0	Файл архива заблокирован
1	80C1	Слишком много открытых файлов: Одновременно может быть открыто восемь файлов архивов данных.
1	8253	Количество наборов данных недействительно
1	8353	Недопустимый выбор формата
1	8453	Недействительный выбор отметки времени
1	8B24	Неправильное назначение области HEADER: Пример: Ссылка на локальную память
1	8B51	Недействительный тип данных для параметра HEADER
1	8B52	Слишком много элементов данных в параметре HEADER
1	8C24	Неправильное назначение области DATA: Пример: Указывает на локальную память
1	8C51	Недействительный тип данных для параметра DATA
1	8C52	Слишком много элементов данных в параметре DATA

DataLogOpen (открыть архив данных)

Таблица 9-203 Инструкция DataLogOpen

LAD/FBD	SCL	Описание

DataLogOpen_DB
DataLogOpen
EN ENO
REQ DONE
MODE BUSY
NAME ERROR
ID STATUS
	`"DataLogOpen_DB"(`
	`req:=_bool_in_,`
	`mode:=_uint_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`name:=_string_inout_,`
	`ID:=_dword_inout_);`
		Открывает уже имеющийся файл архива данных. Необходимо открыть архив данных, чтобы можно было записывать (Страница 518) в него новые наборы данных. Архивы данных могут открываться и закрываться по отдельности. Одновременно может быть открыто до восьми архивов данных.
		При добавлении инструкции, STEP 7 автоматически создает соответствующий DB экземпляра.

[2] В примере SCL "DataLogOpen_DB" - это имя DB экземпляра.

Таблица 9-204 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Инструкцию запускает положительный фронт (0 на 1). (значение по умолчанию: False)
MODE	IN	UInt	Режим работы: • 0 - прикрепить к существующим данным (по умолчанию) • 1 - удалить все существующие наборы данных
NAME	IN	Variant	Имя существующего архива данных: Этот вариант поддерживает только тип данных String и может быть сохранен только в локальной памяти, в DB или M памяти. (значение по умолчанию: ' ')
ID	In/Out	DWord	Числовой идентификатор архива данных. (значение по умолчанию: 0) **Примечание:** Доступ к этому параметру по символьному имени недопустим.
DONE	OUT	Bool	Бит DONE = TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок. (значение по умолчанию: False)
BUSY	OUT	Bool	• 0 - нет обрабатываемых инструкций • 1 - процесс в обработке
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение TRUE, после того, как последний запрос был завершен с ошибкой. Код ошибки в параметре STATUS действителен только в цикле, где ERROR = TRUE.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

Можно ввести NAME или ID (параметрID как вход) уже имеющегося архива данных. Если указываются оба параметра и действительный ID соответствует архиву данных NAME, то используется ID, а NAME игнорируется.

Параметр NAME должен соответствовать имени созданного с помощью инструкции DataLogCreate архива данных. Если указывается только NAME, и NAME ссылается на действительный архив данных, то выводится соответствующий ID (параметр ID как выход).

Примечание

Общее использование файлов архива данных

- После выполнения инструкций DataLogCreate и DataLogNewFile автоматический открываются файлы архива данных.
- После перехода устройства PLC из RUN в STOP или после перезапуска устройства PLC, файлы архива данных закрываются автоматически.
- Для выполнения новой инструкции DataLogWrite, файл архива данных должен быть открыт.
- Одновременно может быть открыто восемь файлов архивов данных. Может существовать больше восьми файлов архива данных, но несколько из них должно быть закрыто, чтобы одновременно было открыто не более восьми файлов.

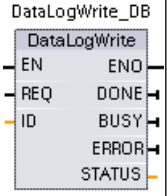
Таблица 9-205 Значения ERROR иSTATUS

ERROR	STATUS (W#16#....)	Описание
0	0000	Ошибки отсутствуют
0	0002	Предупреждение: Файл архива данных уже был открыт этой программой пользователя
0	7000	Вызов без REQ фронта: BUSY = 0, DONE = 0
0	7001	Первый вызов с REQ фронтом (в обработке): BUSY = 1, DONE = 0
0	7002	N-й вызов (в обработке): BUSY = 1, DONE = 0
1	8070	Общая внутренняя экземплярная память занята.
1	8090	Определение архива данных отличается от существующего файла архива данных.
1	8091	Параметр имени не является строкой.
1	8092	Архив данных отсутствует.
1	80C0	Файл архива данных заблокирован.
1	80C1	Слишком много открытых файлов: Одновременно может быть открыто восемь файлов архивов данных.

DataLogWrite (запись архива данных)

Таблица 9-206 Инструкция DataLogWrite

LAD/FBD	SCL	Описание

	`"DataLogWrite_DB"(`
	`req:=_bool_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`ID:=_dword_inout_);`
		Записывает набор данных в указанный архив данных. Уже существующий целевой архив данных должен быть открыт (Страница 516), чтобы с помощью инструкции DataLogWrite можно было бы выполнять запись в архив.
		При добавлении инструкции, STEP 7 автоматически создает соответствующий DB экземпляра.

[2] В примере SCL "DataLogWrite_DB" - это имя DB экземпляра.

Таблица 9-207 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Инструкцию запускает положительный фронт (0 на 1). (значение по умолчанию: False)
ID	In/Out	DWord	Цифровой идентификатор архива данных: Для инструкции DataLogWrite используется только как вход. (значение по умолчанию: 0) **Примечание:** Доступ к этому параметру по символьному имени недопустим.
DONE	OUT	Bool	Бит DONE = TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок.

Параметр и тип данных		Тип данных	Описание
BUSY	OUT	Bool	• 0 - нет обрабатываемых инструкций • 1 - Инструкция обрабатывается
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение TRUE, после того, как последний запрос был завершен с ошибкой. Код ошибки в параметре STATUS действителен только в цикле, где ERROR = TRUE.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

Параметр DATA инструкции DataLogCreate определяет адрес памяти и структуры данных буфера наборов данных. Для передачи новых данных наборов данных из буфера в архив данных, управляющая программа должна загрузить буфер наборов данных с текущими технологическими значениями рабочего цикла и после выполнить инструкцию DataLogWrite.

Параметр ID задает конфигурацию архива данных и набора данных. Инструкция DataLogCreate создает число ID.

Если в круговом файле архива данных есть пустые наборы данных, то инструкция DataLogWrite выполняет запись в следующий свободный набор данных. Если все наборы данных заняты, то инструкция DataLogWrite заменяет самый старый набор данных.

ВНИМАНИЕ

Архив данных должен быть создан до момента запуска инструкции для записи архива данных

Операции DataLogCreate и DataLogNewFile для создания архивов данных растягиваются на несколько программных циклов. Фактически необходимое для создания файла архива время зависит от структуры набора данных и количества наборов данных. Логика программы должна отслеживать и регистрировать переключение бита DONE на TRUE, что является сигналом завершения создания для файла архива. Если инструкция DataLogWrite будет выполнена до завершения создания архива данных, то операция записи не записывает новый архивный набор данных.

Примечание

Влияние архивов данных на внутреннюю память CPU

Каждый процесс записи архива данных использует как минимум 2 кБ памяти. Если программа выполняет частые записи небольшого количества данных, то каждая из записей будет занимать не менее 2 кБ памяти. Лучше собирать небольшие объемы данных в блоках данных (DB), а затем блоки данных записывать в архив данных с менее частыми интервалами.

Если программа выполняет множество записей в архив данных с высокой частотой, то необходимо рассмотреть вопрос об использовании сменной карты памяти SD.

ВНИМАНИЕ

Потенциальная потеря архива данных из-за сбоя питания CPU

Если во время выполнения инструкции DataLogWrite произойдет сбой по питанию, то набор данных, передаваемый в архив данных, может быть потерян.

Таблица 9-208 Значения ERROR иSTATUS

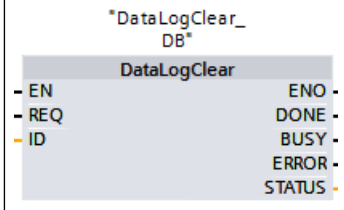
ERROR	STATUS (W#16#....)	Описание
0	0000	Ошибки отсутствуют
0	0001	Показывает, что архив данных заполнен: Каждый архив данных создается с заданным максимальным числом наборов данных. Последний набор данных из этого максимального числа был записан. Следующая инструкция записи заменит самый старый набор данных.
0	7000	Вызов без REQ фронта: BUSY = 0, DONE = 0
0	7001	Первый вызов с REQ фронтом (в обработке): BUSY = 1, DONE = 0
0	7002	N-й вызов (в обработке): BUSY = 1, DONE = 0
1	8070	Общая внутренняя экземплярная память занята.
1	8092	Архив данных отсутствует.
1	80B0	Файл архива данных не открыт (только для режима с явным открытием).

DataLogClear (очистка архива данных)

Описание

Таблица 9-209 Инструкция DataLogClear

LAD/FBD	SCL	Описание

	`"DataLogClear_DB"(`
	`REQ:=_bool_in_,`
	`DONE=>_bool_out_,`
	`BUSY=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`ID:=_dword_inout_);`
		Инструкция "DataLogClear" удаляет все наборы данных из существующего архива данных. Инструкция не удаляет опциональный заголовок файла CSV (см. описание параметра HEADER инструкции "DataLogCreate (Страница 512)").
		С помощью параметра ID выбирается архив данных, наборы данных которого должны быть удалены.

"DataLogClear_DB" - это имя DB экземпляра.

Необходимое условие

Для удаления наборов данных, архив данных должен быть открыт. Для открытия архива данных следует использовать инструкцию DataLogOpen (Страница 516).

Параметр

В приведенной ниже таблице показаны параметры инструкции "DataLogClear":

Параметр	Объявление	Тип данных	Область памяти	Описание
REQ	Input	BOOL	I, Q, M, L, D, T, C или константа (T и C доступны только для S7-1500 в LAD и FBD)	Выполнение инструкции при положительном фронте
ID	InOut	DWORD	I, Q, M, D, L	Цифровой идентификатор архива данных
DONE	Output	BOOL	I, Q, M, D, L	Инструкция выполнена успешно.
BUSY	Output	BOOL	I, Q, M, D, L	Выполнение инструкции еще не завершено.
ERROR	Output	BOOL	I, Q, M, D, L	• 0: Ошибки отсутствуют. • 1: Во время выполнения инструкции произошла ошибка. Подробная информация выводится в параметре STATUS.
STATUS	Output	WORD	I, Q, M, D, L	Параметр состояния Параметр устанавливается только на время одного вызова. Поэтому для отображения состояния параметр STATUS необходимо скопировать в свободную область данных.

Дополнительную информацию о допустимых типах данных можно найти в разделе "Типы данных (Страница 123)".

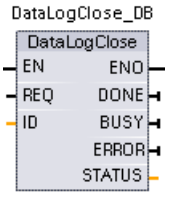
Параметр STATUS

Код ошибки* (W#16#...)	Значение
0000	Ошибки отсутствуют.
7000	Нет активных обрабатываемых заданий.
7001	Запуск обработки задания. Параметр BUSY = 1, DONE = 0
7002	Промежуточный вызов (REQ не играет роли): Инструкция уже активна; BUSY имеет значение 1.
8080	Выбранный с помощью параметра ID файл архива данных не может быть обработан с помощью инструкции "DataLogClear".
8092	Архив данных отсутствует.
80A2	Возвращаемая файловой системой ошибка записи.
80B0	Архив данных не открыт.
80B4	Карта памяти имеет защиту от записи.
* Коды ошибок могут отображаться в редакторе текстов программ в виде целых или шестнадцатеричных значений. Информацию о переключении форматов отображения см. "См. также".	

DataLogClose (закрытие архива данных)

Таблица 9-210 Инструкция DataLogClose

LAD/FBD	SCL	Описание

	`"DataLogClose_DB"(`
	`req:=_bool_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`ID:=_dword_inout_);`
		Закрывает открытый файл архива данных. Выполненная для закрытого архива данных инструкция DataLogWrite вызывает ошибку. Инструкции записи для этого архива данных возможны только после выполнения инструкции DataLogOpen.
		При переходе в рабочее состояние STOP все открытые файлы архива данных закрываются.
		При добавлении инструкции, STEP 7 автоматически создает соответствующий DB экземпляра.

[2] В примере SCL "DataLogClose_DB" - это имя DB экземпляра.

Таблица 9-211 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Инструкцию запускает положительный фронт (0 на 1). (значение по умолчанию: False)
ID	In/Out	DWord	Числовой идентификатор архива данных. Для инструкции DataLogClose используется только как вход. (значение по умолчанию: 0) **Примечание:** Доступ к этому параметру по символьному имени недопустим.
DONE	OUT	Bool	Бит DONE = TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок.
BUSY	OUT	Bool	• 0 - нет обрабатываемых инструкций • 1 - процесс в обработке
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение TRUE, после того, как последний запрос был завершен с ошибкой. Код ошибки в параметре STATUS действителен только в цикле, где ERROR = TRUE.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

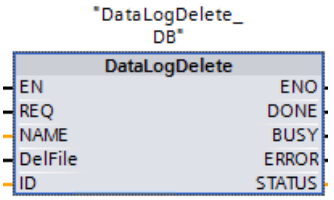
Таблица 9-212 Значения ERROR иSTATUS

ERROR	STATUS (W#16#....)	Описание
0	0000	Ошибки отсутствуют
0	0001	Архив данных не открыт
0	7000	Вызов без REQ фронта: BUSY = 0, DONE = 0
0	7001	Первый вызов с REQ фронтом (в обработке): BUSY = 1, DONE = 0
0	7002	N-й вызов (в обработке): BUSY = 1, DONE = 0
1	8092	Архив данных отсутствует.

DataLogDelete (удаление архива данных)

Таблица 9-213 Инструкция DataLogDelete

LAD/FBD	SCL	Описание

	`"DataLogDelete_DB"(`
	`REQ:=_bool_in_,`
	`NAME:=_variant_in_,`
	`DelFile:=_bool_in_,`
	`DONE=>_bool_out_,`
	`BUSY=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`ID:=_dword_inout_);`
		С помощью инструкции "DataLogDelete" можно удалить файл архива данных. Архив данных и содержащиеся в нем наборы данных могут быть удалены только в том случае, если они были созданы с помощью инструкции "DataLogCreate" или "DataLogNewFile".

"DataLogDelete_DB" - это имя DB экземпляра.

Параметр

В приведенной ниже таблице показаны параметры инструкции "DataLogDelete":

Параметр	Объявление	Тип данных	Область памяти	Описание
REQ	Input	BOOL	I, Q, M, L, D, T, C или константа (T и C доступны только для S7-1500 в LAD и FBD)	Выполнение инструкции при положительном фронте
NAME	Input	VARIANT	L, D	Имя файла архива данных
DELFILE	Input	BOOL	I, Q, M, D, L или константа	• 0: Архив данных будет сохранен. • 1: Архив данных будет удален.
ID	InOut	DWORD	I, Q, M, D, L	Цифровой идентификатор архива данных
DONE	Output	BOOL	I, Q, M, D, L	Инструкция выполнена успешно.
BUSY	Output	BOOL	I, Q, M, D, L	Удаление архива данных еще не завершено.
ERROR	Output	BOOL	I, Q, M, D, L	• 0: Ошибки отсутствуют. • 1: Во время выполнения инструкции произошла ошибка. Подробная информация выводится в параметре STATUS.
STATUS	Output	WORD	I, Q, M, D, L	Параметр состояния Параметр устанавливается только на время одного вызова. Поэтому для отображения состояния параметр STATUS необходимо скопировать в свободную область данных.

Дополнительную информацию о допустимых типах данных можно найти в разделе "Типы данных (Страница 123)".

Параметры NAME иID

Выбрать архив данных, подлежащий удалению, с помощью параметров NAME и ID. Параметр ID обрабатывается в первую очередь. Если существует архив данных со значащим ID, то параметр NAME не обрабатывается. Если в параметре ID используется значение 0, то в параметреNAME должно использоваться значение с типом данных STRING.

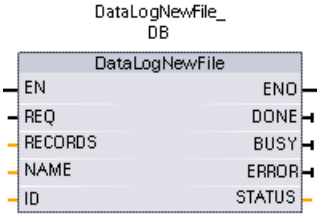
Параметр RET_VAL

Код ошибки* (W#16#...)	**Значение**
0	Ошибки отсутствуют.
7000	Нет активных обрабатываемых заданий.
7001	Запуск обработки задания. Параметр BUSY = 1, DONE = 0
7002	Промежуточный вызов (REQ не играет роли): Инструкция уже активна; BUSY имеет значение 1.
8091	В параметре NAME используется тип данных, отличающийся от STRING.
8092	Архив данных отсутствует.
80A2	Возвращаемая файловой системой ошибка записи.
80B4	Карта памяти имеет защиту от записи.
* Коды ошибок могут отображаться в редакторе текстов программ в виде целых или шестнадцатеричных значений. Информацию о переключении форматов отображения см. "См. также".	

DataLogNewFile (архив данных в новом файле)

Таблица 9-214 Инструкция DataLogNewFile

LAD/FBD	**SCL**	**Описание**

	`"DataLogNewFile_DB"(`
	`req:=_bool_in_,`
	`records=:_udint_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`name=:_DataLog_out_,`
	`ID:=_dword_inout_);`
		Позволяет программе создать новый файл архива данных на основе существующего файла архива данных.
		При добавлении инструкции, STEP 7 автоматически создает соответствующий DB экземпляра.

[2] В примере SCL "DataLogNewFile_DB" - это имя DB экземпляра.

Таблица 9-215 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Инструкцию запускает положительный фронт (0 на 1). (значение по умолчанию: False)
RECORDS	IN	UDInt	Максимальное количество наборов данных, которое может содержать круговой архив данных перед заменой самой старой записи. (значение по умолчанию: 1) Набор данных с заголовком не входит в это число. Для успешного создания архива данных необходим достаточный объем загружаемой памяти CPU.
NAME	IN	Variant	Имя архива данных: Вводится имя. Этот вариант поддерживает только тип данных String и может быть сохранен только в локальной памяти, в DB или M памяти. (значение по умолчанию: ' ') Эта символьная строка используется и для имени файла архива данных. Содержащиеся в имени символы должны отвечать правилам для присвоения имен в файловой системе Windows. Символы \ / : * ? " < > | и пробел запрещены.
ID	In/Out	DWord	Цифровой идентификатор архива данных (значение по умолчанию: 0): • При выполнении вход ID указывает действительный архив данных. Новая конфигурация архива данных копируется из этого архива данных. • После выполнения параметр ID становится выходом, выводящим ID нового созданного файла архива данных. **Примечание:** Доступ к этому параметру по символьному имени недопустим.
DONE	OUT	Bool	Бит DONE = TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок.
BUSY	OUT	Bool	• 0 - нет обрабатываемых инструкций • 1 - процесс в обработке
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение TRUE, после того, как последний запрос был завершен с ошибкой. Код ошибки в параметре STATUS действителен только в цикле, где ERROR = TRUE.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

Инструкция DataLogNewFile может быть выполнена тогда, когда архив данных заполнен или почти заполнен и сохраненные в нем данные не должны быть потеряны. На основе структуры заполненного файла архива данных может быть создан новый, пустой файл архива данных. Запись-заголовок дублирует свойства исходного архива данных (установки DATA для буфера наборов данных, формат данных и отметка времени) из исходного архива данных. Исходный файл архива данных не явно закрывается, а новый файл архива данных не явно открывается.

Запуск DataLogWrite через параметры: Программа должна контролировать параметры ERROR и STATUS каждой инструкции DataLogWrite. Если последний набор данных записан и архив данных заполнен, то бит ERROR инструкции DataLogWrite =1 и слово данных STATUS инструкции DataLogWrite = 1. Эти значения ERROR и STATUS действительный только в течение одного цикла, поэтому логика контроля должна использовать ERROR = 1 в качестве временного окна для регистрации значения STATUS с последующей проверкой на предмет STATUS = 1 (архив данных заполнен).

Принцип работы DataLogNewFile: После получения логикой программы сигнала "Архив данных заполнен", в этом состоянии активируется инструкция DataLogNewFile. Инструкция DataLogNewFile должна быть выполнена с ID существующего (обычно заполненного) и открытого архива данных, но параметр NAME должен быть новым и однозначным. После выполнения инструкции DataLogNewFile выводится новое значение ID для архива данных (как выходной параметр), соответствующее имени нового архива данных. Новый файл архива данных не явно открывается и в него могут сохраняться новые наборы данных. Новые инструкции DataLogWrite, обращающиеся к новому файлу архива данных, должны использовать выведенное инструкцией DataLogNewFile значение ID.

ВНИМАНИЕ
Архив данных должен быть создан до момента запуска инструкции для записи архива данных Операции DataLogCreate и DataLogNewFile для создания архивов данных растягиваются на несколько программных циклов. Фактически необходимое для создания файла архива время зависит от структуры набора данных и количества наборов данных. Логика программы должна отслеживать и регистрировать переключение бита DONE на TRUE, что является сигналом завершения создания для файла архива. Если программа пользователя выполнит инструкцию DataLogWrite до завершения создания архива данных, то операция записи нового архивного набора данных может пройти не так, как положено.

Таблица 9-216 Значения ERROR иSTATUS

ERROR	STATUS (W#16#....)	Описание
0	0000	Ошибки отсутствуют
0	7000	Вызов без REQ фронта: BUSY = 0, DONE = 0
0	7001	Первый вызов с REQ фронтом (в обработке): BUSY = 1, DONE = 0
0	7002	$N^{-й}$ вызов (в обработке): BUSY = 1, DONE = 0
1	8070	Общая внутренняя экземплярная память занята.
1	8090	Недействительное имя файла
1	8091	Параметр имени не является строкой.
1	8092	Архив данных отсутствует.
1	8093	Архив данных уже существует.
1	8097	Требуемая длина файла превышает макс. размер файла для файловой системы.
1	80B2	Отсутствуют дополнительные ID ресурсов Примечание: Удалить несколько имеющихся архивов данных для освобождения места для нового архива данных.
1	80B3	Недостаточно свободной загружаемой памяти.
1	80B4	Карта памяти имеет защиту от записи.
1	80C1	Слишком много открытых файлов.

9.9.2.3 Работа с архивами данных

Файлы архива данных сохраняются в постоянной флэш-памяти в формате CSV (разделенные запятой значения). Можно просматривать архивы данных через веб-сервер PLC или вставив карту памяти из устройства PLC в обычный компьютерный кардридер.

Просмотр архивов данных через веб-сервер PLC

Если PROFINET порт устройства PLC и компьютер подключены к сети, то можно использовать обычный веб-браузер, например Microsoft Internet Explorer или Mozilla Firefox, для доступа к встроенному в PLC веб-серверу. При работе с веб-сервером PLC устройство PLC может находиться в режимах RUN или STOP. Если устройство PLC находится в режиме RUN, то управляющая программа продолжает выполняться, даже при передаче архивных данных по сети с помощью веб-сервера PLC.

Доступ к веб-серверу:

1. Активировать веб-сервер в конфигурации устройства для целевого CPU (Страница 1026).
2. Подключить свой PC по сети PROFINET в целевой системе (Страница 1029).
3. Обратиться с помощью встроенного веб-сервера к CPU (Страница 1036).
4. Теперь можно загружать, редактировать и удалять файлы архива данных с помощью стандартной веб-страницы "Браузер файлов" (Страница 1068).
5. Открыть файл .csv с помощью программы обработки электронных таблиц, например, Excel.

Примечание

Управление архивами данных

Хранить не более 1000 архивов данных в файловой системе. При превышении данного значения веб-серверу может не хватить ресурсов CPU для отображения архивов данных.

Если случится, что веб-страница браузера файлов не сможет отобразить архивы данных, то необходимо перевести CPU в режим STOP, чтобы можно было просматривать и удалять архивы данных.

Управление архивами данных должно быть организовано таким образом, чтобы сохранялось только необходимо число архивов данных и чтобы их количество не превышало максимально допустимого значения, равного 1000 записей.

Просмотр архивов данных на карте памяти устройства PLC

Если в S7-1200 CPU вставлена "программная" карта памяти, то можно извлечь эту карту памяти и вставить ее в стандартный слот SD (Secure Digital) или MMC (MultiMediaCard) в PC или PG. Устройство PLC при извлечении карты памяти находится в рабочем состоянии STOP и управляющая программа не выполняется.

Использовать проводник Windows для перехода к папке \DataLog карты памяти. В этой папке находятся все файлы архивов данных (*.csv).

Скопировать файлы архивов данных на локальный жесткий диск PC. Теперь с помощью программы Excel можно открыть локальную копию csv-файла, а не оригинальные файлы, сохраненные на карте памяти.

ВНИМАНИЕ
С помощью картридера можно копировать, но не изменять или удалять файлы архивов данных с карты памяти S7-1200. В качестве рекомендованного инструмента для просмотра, загрузки (копирования) и удаления файлов архива данных можно использовать стандартный браузер файлов веб-сервера. При прямом просмотре содержимого карты памяти в проводнике Windows есть риск случайного удаления или изменения файлов архива данных или других системных файлов, повредив файл или сделав непригодной карту памяти.

ВНИМАНИЕ
Влияние архивов данных на карты памяти Для обеспечения наибольшей производительности и надежности системы, следует ограничить частоту записи данных на карту памяти до 200 мс.

9.9.2.4 Ограничение размера для файлов архивов данных

Файлы архивов данных занимают место в загружаемой памяти PLC, наряду с программами, программными данными, конфигурационными данными, пользовательскими веб-страницами и системными данными PLC. Большой программе необходим большой объем загружаемой памяти. Для файлов архивов данных может не хватить свободной загружаемой памяти. В этом случае можно использовать карту памяти как "программную карту" (Страница 142) для увеличения размера загружаемой памяти. S7-1200 CPU могут использовать внутреннюю или внешнюю загружаемую память, но не обе одновременно.

Условие для максимального размера файлов архивов данных

Максимальный размер одного файла архива данных ограничен свободной загружаемой памятью или 500 мегабайтами (в зависимости от того, что из них меньше). Размер в 500 мегабайт в данном случае имеет отношение к десятичному определению мегабайта, например, поэтому максимальный размер файла архивов данных составляет 500.000.000 байт или 500 x 1000^2 байт.

Таблица 9-217 Размер загружаемой памяти

Область данных	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C, CPU 1217C	Хранение данных
Внутренняя загружаемая память Флэш-память	1 МБ	2 МБ	4 МБ	4 МБ	Программа пользователя и программные данные, конфигурационные данные, архивы данных, пользовательские веб-страницы и системные данные PLC
Внешняя загружаемая память Опциональная карта памяти как "программная карта"	4 МБ, 12 МБ, 24 МБ, 256 МБ, 2 ГБ или 32 ГБ в зависимости от объема размера карты SD				

Определение размера свободной загружаемой памяти

Свободный размер загружаемой памяти изменяется при выполнении различных операций, т.к. операционная система периодически занимает и освобождает память. Для определения размера загружаемой памяти выполнить следующие шаги.

1. Установить онлайн-соединение между STEP 7 и целевой системой S7-1200.
2. Загрузить программу, управляющую инструкциями архивов данных.
3. Создать при необходимости опциональные пользовательские веб-страницы. Стандартные веб-страницы для доступа к файлам архивов данных сохранены в прошивке PLC и не занимают место в загружаемой памяти.
4. Запросить через Интерактивные функции и диагностика (Страница 1387) или страничку диагностики веб-сервера (Страница 1043) общий размер загружаемой памяти и размер свободного места.

Вычисление размера файла архива данных (все наборы данных)

При создании файла архива данных CPU выделяет максимальный объем памяти. В дополнение к объему памяти, необходимой для всех наборов данных, необходимо предусмотреть область памяти для хранения заголовка журнала (если используется), заголовка меток времени (если используются), заголовка индекса записи и минимальный размер блока для распределения памяти.

Для определения размера файлов архивов данных следует использовать приведенную ниже формулу и помнить об ограничении для максимального размера файлов.

Байты данных архива данных = ((байты данных в наборе данных в байтах + байты меток времени + 12 байт) * количество наборов данных)

Header (Заголовок)

Байты заголовка архива данных = символьные байты заголовка + 2 байта

Символьные байты заголовка

- Без заголовка данных и без меток времени = 7 байт
- Без заголовка данных и метки времени (с заголовком меток времени) = 21 байт
- Заголовок данных и без меток времени = число символьных байт во всех заголовках столбцов, включая разделители в виде запятых
- Заголовок данных и метки времени (с заголовком меток времени) = число символьный байт во всех заголовках столбцов, включая разделители в виде запятых + 21 байт

Data (Данные)

Байты данных архива данных = ((байты данных в наборе данных в байтах + байты меток времени + 12 байт) * количество наборов данных)

Байты данных в наборе данных

Параметр DATA инструкции DataLogCreate указывает на структуру, которая определяет количество полей данных и тип данных каждого поля данных для одного архивного набора данных.

Умножить количество вхождений для данного типа данных на количество требуемых байтов. Повторить процесс для каждого типа данных в одном наборе данных и суммировать все байты данных, чтобы получить сумму всех элементов данных в одном наборе данных.

Размер отдельных элементов данных

Данные файлов архива данных сохраняются как символьные байты в формате CSV (разделенные запятой значения). В приведенной ниже таблице показано количество байт, необходимых для сохранения каждого элемента данных.

Тип данных	Количество байт (включая данные плюс один байт для запятой)
Bool	2
Byte	5
Word	7
DWord	12
Char	4

String	**Пример 1**: MyString[10] Максимальный размер символьной строки - 10 символов. Текстовые символы + автоматическое заполнение пустыми символами = 10 байт Двойные кавычки открытия и закрытия + символы запятой = 3 байта 10 + 3 = 13 байт всего **Пример 2**: Mystring2 Если размер не задан в квадратных скобках, то по умолчанию выделяется 254 байт. Текстовые символы + автоматическое заполнение пустыми символами = 254 байт Двойные кавычки открытия и закрытия + символы запятой = 3 байта 254 + 3 = 257 байт всего
USInt	5
UInt	7
UDInt	12
SInt	5
Int	7
DInt	12
Real	16
LReal	25
Time	15
DTL	24

Количество наборов данных в одном файле архива данных

Параметр RECORDS инструкции DataLogCreate устанавливает максимальное количество наборов данных в файле архива данных.

Байты отметок времени в наборе данных

- Без меток времени = 0 байт
- Метка времени = 20 байт

9.9.2.5 Пример программы для архивов данных

Этот пример программы для архивов данных не показывает всю логику программы, необходимую для запроса считываемых значений из динамического процесса, но приведены основные операции, выполняемые инструкциями для архивов данных. Структура и количество используемых архивных файлов зависит от требований к управлению процессом.

Примечание

Общее использование файлов архива данных

- После выполнения инструкций DataLogCreate и DataLogNew автоматический открываются файлы архива данных.
- После перехода устройства PLC из RUN в STOP или после перезапуска устройства PLC, файлы архива данных закрываются автоматически.
- Для выполнения инструкции DataLogWrite, файл архива данных должен быть открыт.
- Одновременно может быть открыто восемь файлов архивов данных. Может существовать больше восьми файлов архива данных, но несколько из них должно быть закрыто, чтобы одновременно было открыто не более восьми файлов.

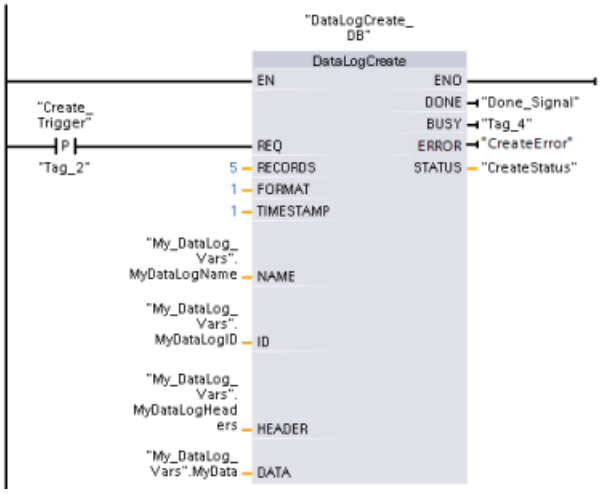
Пример программы для архивов данных

Примеры имен архивов данных, текстовых заголовков и структуры MyData создаются в наборе данных. Три переменные MyData используются для временного хранения новых считанных значений. Считываемые технологические значения по этим адресам DB при выполнении инструкции DataLogWrite передаются в файл архива данных.

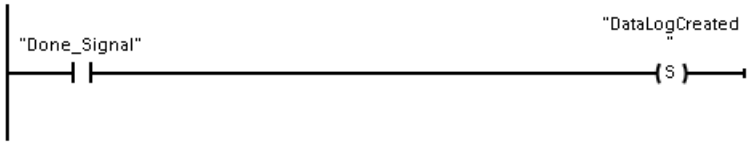
My_Datalog_Vars

	Name	Data type	Start value
1	Static		
2	MyNewDataLogName	String	'MyNEWDataLog'
3	MyDataLogName	String	'MyDataLog'
4	MyDataLogID	DWord	0
5	MyDataLogHeaders	String	'Count,Temperature,Pressure'
6	MyData	Struct	
7	MyCount	Int	0
8	MyTemperature	Real	0.0
9	MyPressure	Real	0.0

Сегмент 1 Положительный фронт на REQ запускает процесс создания архива данных.



Сегмент 2 Снять выход DONE из DataLogCreate, потому что он действителен только в течение одного цикла.



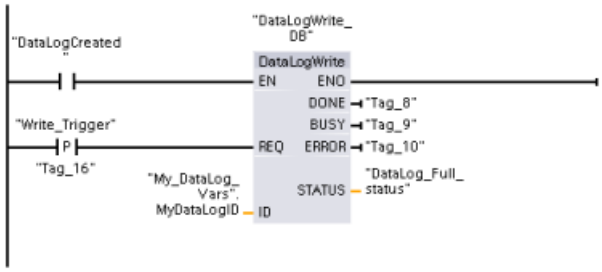
Сегмент 3 При наличии ошибки сохранить выход состояния.



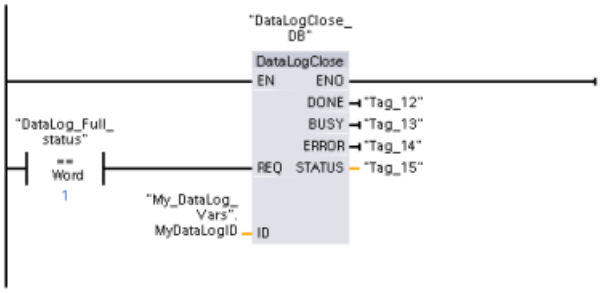
Сегмент 4 Положительный фронт инициирует момент времени для сохранения новых технологических значений в структуре MyData.



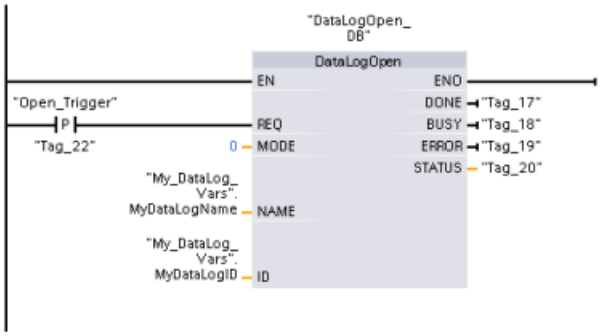
Сегмент 5 Состояние входа EN использует момент времени, к которому выполнение DataLogCreate завершено. Процесс создания занимает несколько циклов и должен быть завершен, чтобы мог быть выполнен процесс записи. Положительный фронт на входе REQ - это событие, запускающее активную операцию записи.



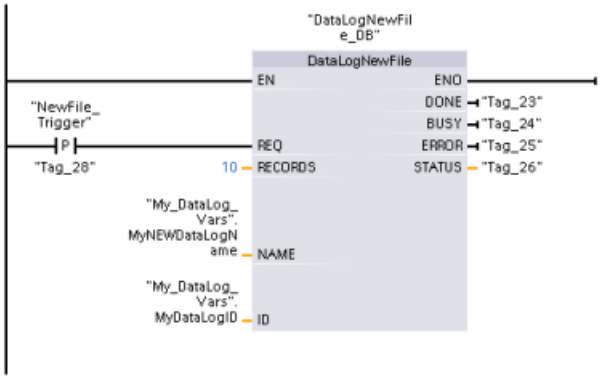
Сегмент 6 Закрыть архив данных после записи последнего набора данных. После выполнения инструкции DataLogWrite, записывающей последний набор данных, сигнализируется состояние "Архивный файл заполнен", если выход STATUS из DataLogWrite: = 1.



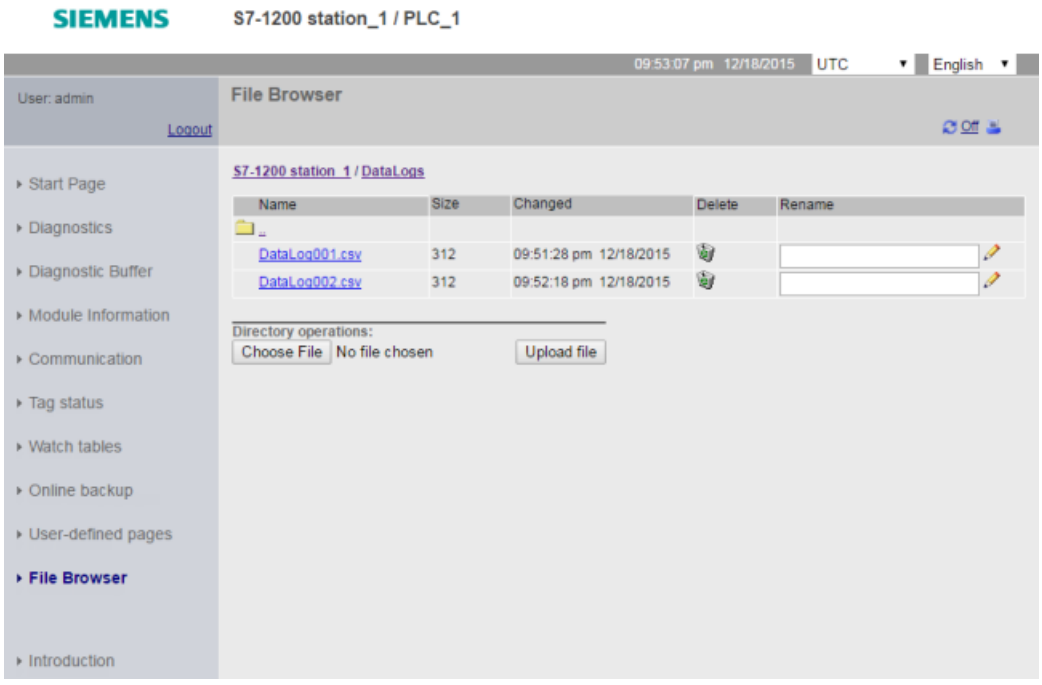
Сегмент 7 Положительный фронт на входе REQ инструкции DataLogOpen моделирует процесс, при котором пользователь нажимает на устройстве HMI кнопку, открывающую файл архива данных. При открытии файла архива данных, в котором все наборы данных заняты технологическими данными, инструкция DataLogWrite при следующем выполнении перезаписывает самый старый набор данных. Но можно сохранить старый архив данных, создав вместо него новый архив данных. Это будет показано в сегменте 7.



Сегмент 8 ID параметр имеет тип IN/OUT. Сначала указывается значение ID существующего архива данных, структура которого должна быть скопирована. После выполнения инструкции DataLogNewFile, на адрес ID ссылки записывается новое и однозначное значение ID для нового архива данных. Требуемая регистрация Бит ONE = TRUE не отображается. Пример для логики битов DONE можно посмотреть в сегментах 1, 2 и 4.



Созданные в примере программы и отображаемые с помощью веб-сервера S7-1200 CPU файлы архива данных



① Опция "Delete" (Удалить) доступна только при наличии прав на редактирование.

② Опция "Rename" (Переименовать) доступна только при наличии прав на редактирование.

Таблица 9-218 Отображаемые в Excel, загруженные примеры для файлов CSV

Два записанных наборов данных в одном файле, который может содержать до пяти наборов данных

	A	B	C	D	E	F
1	Record	Date	UTC Time	Count	Temperature	Pressure
2	1	9/29/2010	21:01:46	5	5.00E+00	5.00E+00
3	2	9/29/2010	21:01:47	5	5.00E+00	5.00E+00
4						
5						

Пять наборов данных в одном файле архива данных, который может содержать до пяти наборов данных

	A	B	C	D	E	F
1	Record	Date	UTC Time	Count	Temperature	Pressure
2	1	9/30/2010	20:26:56	1	9.86E+01	3.52E+01
3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01
4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01
5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01
6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01
7						

После записи в изображенный выше файл, который уже заполнен, еще одного набора данных, шестой процесс записи заменяет самый старый набор данных 1 набором данных 6. Следующий процесс записи заменяет набор данных 2 набором данных 7 и т. д.

	A	B	C	D	E	F
1	Record	Date	UTC Time	Count	Temperature	Pressure
2	6	9/30/2010	20:32:03	6	9.86E+01	3.58E+01
3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01
4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01
5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01
6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01
7						

Примечание

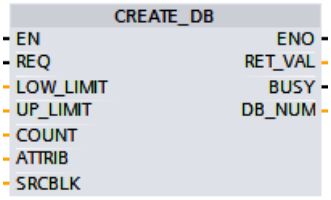
Метки //END больше не устанавливаются на конце не заполненного архива данных. До S7-1200 CPU V4.1 не заполненные до конца архивы данных содержали метку //END.

9.10 Управление блоком данных

9.10.1 CREATE_DB (создание блока данных)

Таблица 9-219 Инструкция CREATE_DB

LAD/FBD	SCL	Описание

	`ret_val := CREATE_DB(`
	`REQ:=_bool_in_,`
	`LOW_LIMIT:=_uint_in_,`
	`UP_LIMIT:=_uint_in_,`
	`COUNT:=_udint_in_,`
	`ATTRIB:=_byte_in_,`
	`BUSY=>_bool_out_,`
	`DB_NUM=>_uint_out_);`
		С помощью инструкции "CREATE_DB" можно создать новый блок данных в загружаемой и/или рабочей памяти.
		Инструкция "CREATE_DB" не изменяет контрольную сумму программы пользователя.
		Блок данных, создаваемый только в рабочей памяти, имеет следующие свойства:
		• После восстановления исходного состояния памяти или ВЫКЛЮЧИТЬ / ВКЛЮЧИТЬ этот блок удаляется.
		• При загрузке или переходе из STOP в RUN содержание не изменяется.

Номер блока данных

Созданному блоку данных присваивается номер из диапазона, определенного параметрами LOW_LIMIT (нижнее предельное значение) и UP_LIMIT (верхнее предельное значение). "CREATE_DB" присваивает DB наименьший возможный номер из указанного диапазона. Номера уже имеющихся в программе пользователя блоков данных присваивать нельзя.

Для создания DB с определенным номером необходимо ввести один и тот же номер для верхнего и нижнего предельного значения указанного диапазона. Если блок данных с таким номером уже существует в рабочей и/или загружаемой памяти или блок данных существует в виде скопированного варианта, инструкция завершается, и в параметре RET_VAL выводится сообщение об ошибке.

Стартовые значения блока данных

С помощью параметра SRCBLK определяются стартовые значения для создаваемого DB. Параметр SRCBLK является указателем на блок данных или область блока данных, из которой берутся стартовые значения. Блок данных, адресуемый в параметре SRCBLK, должен быть создан с доступом по умолчанию (атрибут "Оптимизированный доступ к блоку данных" деактивирован).

- Если область, указанная в параметре SRCBLK, больше, чем созданный блок данных, в качестве стартовых значений принимаются значения до длины созданного блока данных.
- Если область, указанная в параметре SRCBLK, меньше, чем созданный блок данных, остальные значения заполняются "0".

Для обеспечения целостности данных запрещается изменять эту область данных во время обработки "CREATE_DB" (т. е. до тех пор, пока параметр BUSY имеет значение TRUE).

Описание функции

Инструкция "CREATE_DB" работает асинхронно. Обработка растягивается на несколько вызовов. Для запуска задания используется вызов "CREATE_DB" с REQ = 1.

Выходные параметры RET_VAL и BUSY сообщают о состоянии задания.

См. также: DELETE_DB (удаление блока данных) (Страница 546)

Параметры

В приведенной ниже таблице показаны параметры инструкции "CREATE_DB":

Параметры	Объявление	Тип данных	Область памяти	Описание
REQ	Input	BOOL	I, Q, M, D, L или константа	Запускаемый по уровню управляющий параметр "request to activate" REQ = 1: Запрос на создание блока данных
LOW_LIMIT	Input	UINT	I, Q, M, D, L или константа	Нижнее предельное значение диапазона для назначения номера DB. Наименьший возможный номер 60000.
UP_LIMIT	Input	UINT	I, Q, M, D, L или константа	Верхнее предельное значение используемого "CREATE_DB" диапазона для назначения номера DB (максимально возможный номер DB: 60999)
COUNT	Input	UDINT	I, Q, M, D, L или константа	Числовое значение указывает количество байт, которое требуется сгенерировать для созданного блока данных. Для количества байт должен быть указан четный номер. Максимальная длина составляет 65534 байт.

<table>
<tr><th>Параметры</th><th>Объявление</th><th>Тип данных</th><th>Область памяти</th><th>Описание</th></tr>
<tr><td rowspan="8">ATTRIB</td><td rowspan="8">Input</td><td rowspan="8">BYTE</td><td rowspan="8">I, Q, M, D, L или константа</td><td>С помощью первых 4 бит байта в параметре ATTRIB можно определить свойства блока данных*:</td></tr>
<tr><td>• Бит 0 = 0: Атрибут "Сохранять только в загружаемой памяти" не определен.
• Бит 0 = 1: Атрибут "Сохранять только в загружаемой памяти" определен. При этой установке DB не занимает место в рабочей памяти и не включается в программу. Обращение к DB через битовые команды невозможно. Если бит 0 = 1, то выбор для бит 2 не важен.
Для обеспечения совместимости со STEP 7 V5.x, биты 0 и 3 должны учитываться совместно (см. ниже).</td></tr>
<tr><td>• Бит 1 = 0: Атрибут "Блок данных защищен от записи в устройстве" не установлен.
• Бит 1 = 1: Атрибут "Блок данных защищен от записи в устройстве" установлен.</td></tr>
<tr><td>• Бит 2 = 0: Сохраняющийся DB (только для блоков данных, созданных в загружаемой и рабочей памяти). DB считается сохраняющимся, если как минимум одно значение было определено как сохраняющееся.
• Бит 2 = 1: DB не является сохраняющимся (retentive).
Сохранение не распространяется на DB, которые находятся только в загружаемой памяти или только в рабочей памяти. Если инструкция "CREATE_DB" вызывается с одной из двух комбинаций "с сохранением и только в загружаемой памяти" или "с сохранением и только в рабочей памяти", то создаваемый DB не маркируется как сохраняющийся (retentive).</td></tr>
<tr><td>• Бит 3 = 0: Создание DB либо в загружаемой памяти, либо в рабочей памяти (выбор через бит 0, см. выше)
• Бит 3 = 1: Создание блока данных как в загружаемой памяти, так и в рабочей памяти (бит 0 не имеет значения)
Для обеспечения совместимости со STEP 7 V5.x, биты 0 и 3 должны использоваться совместно. Если бит 3 = 1, то выбор бит 0 не имеет значения.</td></tr>
<tr><td><table>
<tr><th>Бит 0</th><th>Бит 3</th><th>Создание DB</th></tr>
<tr><td>0</td><td>0</td><td>Только в рабочей памяти</td></tr>
<tr><td>1</td><td>0</td><td>Только в загружаемой памяти</td></tr>
<tr><td>Не имеет значения</td><td>1</td><td>Рабочая и загружаемая память</td></tr>
</table></td></tr>
<tr><td>• Бит 4 = 0 - Стартовые значения не указаны (входные значения в параметре SRCBLK игнорируются).
• Бит 4 = 1 - Стартовые значения указаны (значения соответствуют блоку данных, адресованному с помощью параметра SRCBLK).</td></tr>
</table>

Параметры	Объявление	Тип данных	Область памяти	Описание
SRCBLK	Input	VARIANT	D	Указатель на блок данных, значения которого используются для инициализации создаваемого блока данных.
RET_VAL	Return	INT	I, Q, M, D, L	Информация об ошибке
BUSY	Output	BOOL	I, Q, M, D, L	BUSY = 1: Процесс еще не завершен.
DB_NUM	Output	DB_DYN (UINT)	I, Q, M, D, L	Номер созданного DB.
* Выбранные здесь свойства соответствуют атрибутам в свойствах блока данных.				

Дополнительную информацию о допустимых типах данных можно найти в разделе "Типы данных (Страница 123)".

Параметр RET_VAL

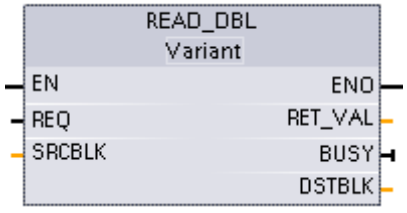
Код ошибки* (W#16#...)	Описание
0000	Ошибки отсутствуют
0081	Целевая область больше исходной области. Исходная область целиком включена в целевую область. Оставшиеся байты целевой области не изменяются.
7000	Первый вызов с REQ = 0: Нет активной передачи данных; BUSY имеет значение 0.
7001	Первый вызов с REQ = 1: Передача данных запущена; BUSY имеет значение 1.
7002	Промежуточный вызов (REQ не играет роли): Передача данных уже активна; BUSY имеет значение 1.
8081	Исходная область больше целевой области. Запись выполняется до заполнения целевой области. Остальные байты исходной области игнорируются.
8092	Функция "Создание блока данных" в настоящий момент недоступна, так как • В настоящее время активна функция "Сжатие памяти пользователя". • Уже достигнуто максимальное количество блоков данных для CPU.
8093	Для параметра SRCBLK не был указан блок данных или указанный блок данных отсутствует в рабочей памяти.
8094	В параметре ATTRIB было указано недопустимое значение.
80A1	Номер ошибки DB: • Номер равен 0. • Номер превышает установленное для CPU верхнее предельное значение для номеров DB. • Нижнее предельное значение > Верхнее предельное значение.
80A2	Ошибка длины DB: • Длина равна 0. • Для длины введено нечетное число. • Длина больше той длины, которую допускает CPU.
80A3	Блок данных в параметре SRCBLK был создан не с доступом по умолчанию.
80B1	Нет свободных номеров для блоков данных.
80B2	Недостаточно рабочей памяти.
80B4	Карта памяти имеет защиту от записи.

Код ошибки* (W#16#...)	Описание
80BB	Недостаточно загружаемой памяти.
80C3	Уже достигнуто максимальное количество одновременно активных инструкций "CREATE_DB".
Общая информация об ошибках	См. также: Общие коды ошибок для расширенных инструкций (Страница 554)
*Коды ошибок могут отображаться в редакторе текстов программ в виде целочисленных значений или шестнадцатеричных значений.	

9.10.2 Инструкции READ_DBL и WRIT_DBL (чтение/запись блока данных в загружаемой памяти)

Таблица 9-220 Инструкции READ_DBL и WRIT_DBL

LAD/FBD	SCL	Описание

	`READ_DBL(`
	`req:=_bool_in_,`
	`srcblk:=_variant_in_,`
	`busy=>_bool_out_,`
	`dstblk=>_variant_out_);`
		Копирует стартовые значения DB или часть значений из загружаемой памяти в целевой DB в рабочей памяти.
		Содержимое загружаемой памяти при копировании не изменяется.

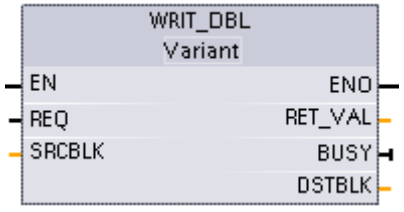
	`WRIT_DBL(`
	`req:=_bool_in_,`
	`srcblk:=_variant_in_,`
	`busy=>_bool_out_,`
	`dstblk=>_variant_out_);`
		Копирует актуальные значения DB или часть значений из рабочей памяти в целевой DB в загружаемой памяти.
		Содержимое рабочей памяти при копировании не изменяется.

Таблица 9-221 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	BOOL	Сигнал с высоким уровнем запускает операцию, если BUSY = 0.
SRCBLK	IN	VARIANT	READ_DBL: Указатель на исходный блок данных в загружаемой памяти WRIT_DBL: Указатель на исходный блок данных в рабочей памяти
RET_VAL	OUT	INT	Условие выполнения
BUSY	OUT	BOOL	BUSY = 1 сигнализирует, что процесс чтения/записи не завершен.
DSTBLK	OUT	VARIANT	READ_DBL: Указатель на целевой блок данных в рабочей памяти WRIT_DBL: Указатель на целевой блок данных в загружаемой памяти

DB обычно создается как в загружаемой (Flash), так и в рабочей памяти (RAM). Стартовые значения (начальные значения) всегда сохраняются в загружаемой памяти, а актуальные значения всегда сохраняются в рабочей памяти. С помощью READ_DBL набор стартовых значений из загружаемой памяти может быть скопирован в текущие значения DB в рабочей памяти, адресованный программой. С помощью инструкции WRIT_DBL можно обновить сохраненные во внутренней загружаемой памяти или на карте памяти текущие значения в рабочей памяти.

Примечание

Влияние инструкцииWRIT_DBL и READ_DBL на флэш-память

Инструкция WRIT_DBL отвечает за процессы записи во флэш-память (во внутреннюю загружаемую память или на карту памяти). Для увеличения срока службы флэш-памяти следует использовать инструкцию WRIT_DBL для сокращения числа обновлений, напр., записей изменений в производственном процессе. По той же причине следует избегать частых вызовов READ_DBL для процессов чтения.

Блоки данных для READ_DBL и WRIT_DBL должны быть созданы до вызова этих инструкций в программе STEP 7. Если исходный DB был создан как блок с типом данных "Стандартный", то целевой DB также должен иметь тип "Стандартный". Если исходный блок данных был создан как тип "С оптимизацией", то целевой блок данных тоже должен иметь тип "С оптимизацией".

Если речь идет о стандартных DB, то может быть указано имя переменной (тега) или значение P#. С помощью значения P# может быть указано и скопировано любое число элементов указанного размера (Byte, Word или DWord). Там может быть скопирована часть или весь DB. Если речь идет об оптимизированных DB, может быть указано только имя переменной (тега). Использование оператора P# невозможно. Если для стандартного DB или оптимизированного DB (или для других типов рабочей памяти) указывается имя переменной (тега), то инструкция копирует данные, на которые ссылается это имя переменной (тега). Здесь речь может идти о пользовательском типе, массиве или базовом элементе. Если речь идет о стандартном DB, а не об оптимизированном DB, то для этих инструкций можно использовать только тип данных Struct. Следует использовать пользовательский тип данных (UDT), если речь идет о структуре в оптимизированной памяти. Только пользовательский тип гарантирует совпадение "типов данных" как для исходной, так и для целевой структуры.

Примечание

Использование структуры (тип данных Struct) в "оптимизированном" DB

При использовании типа данных Struct с "оптимизированными" DB, сначала необходимо создать пользовательский тип данных (UDT) для типа данных Struct. После сконфигурировать исходный и целевой DB с типом данных UDT. Тип данных UDT гарантирует согласованность типов данных в типе данных Struct для обоих DB.

Для "стандартных" DB следует использовать тип данных Struct, не создавая тип данных UDT.

READ_DBL и WRIT_DBL выполняются асинхронно с программным циклом. Обработка растягивается на несколько вызовов READ_DBL и WRIT_DBL. Задание на передачу DB

запускается через вызов REQ = 1 и после выполняется мониторинг выходов BUSY и RET_VAL, чтобы определить, когда передача данных будет завершена и есть ли ошибки.

Примечание

Влияние инструкции WRIT_DBL и READ_DBL на коммуникационную нагрузку

Если инструкция WRIT_DBL или READ_DBL активна постоянно, то потребление этими инструкциями коммуникационных ресурсов может привести к потере соединения STEP 7 с CPU. Поэтом следует использовать для параметра REQ положительный фронт на входе (Страница 224), а не NO- или NC-вход (Страница 219), который может оставаться включенным в течение нескольких циклов (высокий уровень сигала).

Для обеспечения целостности данных, не следует изменять целевую область при обработке READ_DBL или исходную область при обработке WRIT_DBL (т.е. пока параметр BUSY = TRUE).

Ограничения для параметров SRCBLK и DSTBLK:

- Блок данных должен быть предварительно создан, прежде чем на него можно будет ссылаться.
- В указателе VARIANT типа BOOL длина должна делиться на 8.
- В указателе VARIANT типа STRING длина должна совпадать в исходном и целевом указателях.

Информация о рецептурах и настройке оборудования

С помощью инструкции READ_DBL и WRIT_DBL можно управлять рецептурами или информацией о настройке оборудования. По сути, это еще один метод для получения сохраняющихся данных для значений, которые изменяются не очень часто. Мы же хотели ограничить количество операций записи, чтобы предотвратить преждевременное старение флэш-памяти. Это позволит увеличить объем сохраняемой памяти сверх лимита, задаваемого поддерживаемыми данными, по крайней мере для значений, которые изменяются не часто. Можно сохранить рецептурную информацию или информацию о настройке оборудования из рабочей памяти в загружаемую память, используя инструкцию WRIT_DBL, и можно снова извлечь эту информацию из загружаемой памяти в рабочую память, используя инструкцию READ_DBL.

Таблица 9-222 Коды условий

RET_VAL (W#16#...)	Описание
0000	Ошибки отсутствуют
0081	Предупреждение: Исходная область меньше целевой области. Исходные данные будут полностью скопированы, при этом дополнительные байты в целевой области останутся без изменений.
7000	Вызов с REQ =0: BUSY = 0
7001	Первый вызов с REQ =1 (в обработке): BUSY = 1
7002	N-й вызов (в обработке): BUSY = 1
8051	Неправильный тип блока данных

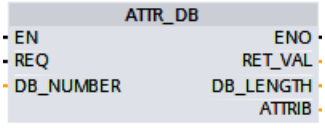
RET_VAL (W#16#...)	Описание
8081	Исходная область больше целевой области. Целевая область заполняется полностью, а оставшиеся байты источника игнорируются.
8251	Неправильный тип блока исходных данных
82B1	Блок исходных данных отсутствует
82C0	Исходный DB обрабатывается другим оператором или функцией коммуникации.
8551	Неправильный тип блока целевых данных
85B1	Блок целевых данных отсутствует
85C0	Целевой DB обрабатывается другим оператором или функцией коммуникации.
80C3	Более 50 операторов READ_DBL или 50 операторов WRIT_DBL в настоящее время находятся в листе ожидания на выполнение.

См. также Рецептуры (Страница 499)

9.10.3 ATTR_DB (чтение атрибутов блока данных)

Таблица 9-223 Инструкция ATTR_DB

LAD/FBD	SCL	Описание

	`ret_val := ATTR_DB(`
	`REQ:=_bool_in_,`
	`DB_NUMBER:=_uint_in_,`
	`DB_LENGTH=>_udint_out_,`
	`ATTRIB=>_byte_out_);`
		С помощью инструкции "ATTR_DB" вызывается информация о блоке данных (DB), находящемся в рабочей памяти CPU. Инструкция определяет установленные в параметре ATTRIB атрибуты для выбранного блока данных.
		Длина не может быть считана для блоков данных с оптимизированным доступом, которые находятся только в загружаемой памяти. В таких случаях параметр DB_LENGH имеет значение 0.
		Не использовать ATTR_DB для блоков данных с оптимизированным доступом и активированным резервом памяти.
		Для чтения блоков данных с управлением перемещением не следует использовать инструкцию "ATTR_DB". Выводится код ошибки 80B2.

Параметр

В приведенной ниже таблице показаны параметры инструкции "ATTR_DB":

Параметр	Объявление	Тип данных	Область памяти	Описание
REQ	Input	BOOL	I, Q, M, D, L или константа	REQ = 1: Запрос на чтение атрибутов блока
DB_NUMBER	Input	DB_ANY	I, Q, M, D, L или константа	Номер тестируемого блока данных
RET_VAL	Output	INT	I, Q, M, D, L	Информация об ошибке
DB_LENGTH	Output	UDINT	I, Q, M, D, L	• Количество байт данных, которое содержится в выбранном блоке данных • 0 для блоков данных с оптимизированным доступом, которые находятся только в загружаемой памяти.
ATTRIB	Output	BYTE	I, Q, M, D, L	Свойства блока данных:
				• Бит 0*= 0: Атрибут "Сохранять только в загружаемой памяти" не определен. • Бит 0*= 1: Атрибут "Сохранять только в загружаемой памяти" определен.
				• Бит 1 = 0: Атрибут "Блок данных защищен от записи в устройстве" не установлен. • Бит 1 = 1: Атрибут "Блок данных защищен от записи в устройстве" установлен.
				Если бит 0 = 1, то бит 2 не важен и получает значение 1. • Бит 2 = 0: С сохранением - DB считается сохраняющимся, если как минимум одно значение было определено как сохраняющееся. • Бит 2 = 1: Без сохранения - Весь DB не является сохраняющимся (retentive).
				• Бит 3*= 0: DB либо в загружаемой памяти (бит 0 = 1), либо в рабочей памяти (бит 0 = 0). • Бит 3*= 1: Блок данных создается как в загружаемой памяти, так и в рабочей памяти.
* Взаимосвязь между битом 0 и битом 3 объясняется в параметрах инструкции "CREATE_DB (создание блока данных) (Страница 537)".				

Дополнительную информацию о допустимых типах данных можно найти в разделе "Типы данных (Страница 123)".

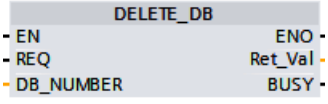
Параметр RET_VAL

Код ошибки* (W#16#...)	Значение
0000	Ошибки отсутствуют.
80A1	Ошибка во входном параметре DB_NUMBER: фактически выбранный параметр • равен 0 • больше максимально допустимого номера DB для используемого CPU
80B1	Блок данных с указанным номером отсутствует в CPU.
80B2	Блоки данных технологических объектов для управления перемещением не могут быть считаны с помощью инструкции "ATTR_DB".
Общая информация об ошибках	См. также: Общие коды ошибок для расширенных инструкций (Страница 554)
*Коды ошибок могут отображаться в редакторе текстов программ в виде целочисленных значений или шестнадцатеричных значений.	

9.10.4 DELETE_DB (удаление блока данных)

Таблица 9-224 Инструкция DELETE_DB

LAD/FBD	SCL	Описание

	`ret_val := DELETE_DB(`
	`  REQ := _bool_in_,`
	`  DB_NUMBER := _uint_in_,`
	`  BUSY => _bool_out_);`
		С помощью инструкции "DELETE_DB" выполняется удаление блока данных (DB), созданного программой пользователя через вызов инструкции "CREATE_DB (Страница 537)".
		Если блок данных был создан не через "CREATE_DB", то DELETE_DB возвращает в параметре RET_VAL код ошибки W#16#80B5.
		Вызов DELETE_DB удаляет выбранный блок данных не сразу же, а после выполнения OB цикла в контрольной точке цикла.

Описание функции

Инструкция "DELETE_DB" работает асинхронно, т.е. ее выполнение растягивается на несколько вызовов. Передача прерывания запускается путем вызова инструкции с REQ = 1.

С помощью выходного параметра BUSY и байтов 2 и 3 выходного параметра RET_VAL отображается состояние задания.

Блок данных полностью удален, если выходной параметр BUSY имеет значение FALSE.

Параметры

В приведенной ниже таблице показаны параметры инструкции "DELETE_DB":

Параметры	Объявление	Тип данных	Область памяти	Описание
REQ	Input	BOOL	I, Q, M, D, L или константа	REQ = 1: Запрос на удаление DB с номером в параметре DB_NUMBER
DB_NUMBER	Input	UINT	I, Q, M, D, L или константа	Номер удаляемого блока данных
RET_VAL	Output	INT	I, Q, M, D, L	Информация об ошибке (см. параметр "RET_VAL").
BUSY	Output	BOOL	I, Q, M, D, L	BUSY = 1: Процесс еще не завершен.

Дополнительную информацию о допустимых типах данных можно найти в разделе "Типы данных (Страница 123)".

Параметр RET_VAL

Код ошибки* (W#16#...)	Значение
0000	Ошибки отсутствуют.
7000	Первый вызов с REQ = 0: Нет активной передачи данных; BUSY имеет значение 0.
7001	Первый вызов с REQ = 1: Передача данных запущена; BUSY имеет значение 1.
7002	Промежуточный вызов (REQ не играет роли): Передача данных уже активна; BUSY имеет значение 1.
80A1	Ошибка во входном параметре DB_NUMBER: • Значение в параметре равно 0. • Значение в параметре больше максимально допустимого номера DB для используемого CPU.
80B1	Блок данных с указанным номером отсутствует в CPU.
80B4	DB не может быть удален, так как карта памяти CPU имеет защиту от записи.
80B5	Блок данных создан не с помощью "CREATE_DB".
80BB	Недостаточно загружаемой памяти.
80C3	Функция "Удалить DB" в настоящее время не может быть выполнена из-за кратковременной нехватки ресурсов.
Общая информация об ошибках	См. также: Общие коды ошибок для расширенных инструкций (Страница 554)
*Коды ошибок могут отображаться в редакторе текстов программ в виде целочисленных значений или шестнадцатеричных значений.	

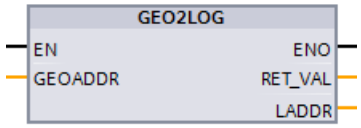
9.11 Обработка адресов

9.11.1 GEO2LOG (определение аппаратного идентификатора из слота)

С помощью GEO2LOG аппаратный идентификатор определяется на основе информации слота.

Таблица 9-225 Инструкция GEO2LOG

LAD/FBD	SCL	Описание

	`ret_val := GEO2LOG(`
	`  GEOADDR:=_variant_in_out_,`
	`  laddr:=_word_out_);`
		С помощью GEO2LOG аппаратный идентификатор определяется на основе информации слота.

С помощью инструкции GEO2LOG аппаратный идентификатор определяется на основе информации слота, устанавливаемой через системный тип данных GEOADDR:

В зависимости от типа оборудования, определенного в параметре HWTYPE, анализируется следующая информация других параметров GEOADDR:

- При HWTYPE = 1 (система PROFINET IO):
  - Анализируется только IOSYSTEM. Другие параметры GEOADDR не учитываются.
  - Выводится аппаратный идентификатор системы PROFINET IO.
- При HWTYPE = 2 (устройство PROFINET IO):
  - Анализируются IOSYSTEM и STATION. Другие параметры GEOADDR не учитываются.
  - Выводится аппаратный идентификатор устройства PROFINET IO.
- При HWTYPE = 3 (стойка):
  - Анализируются только IOSYSTEM и STATION. Другие параметры GEOADDR не учитываются.
  - Выводится аппаратный идентификатор монтажной стойки.
- При HWTYPE = 4 (модуль):
  - Анализируются IOSYSTEM, STATION, и SLOT. Параметр SUBSLOT из GEOADDR не учитывается.
  - Выводится аппаратный идентификатор модуля.
- При HWTYPE = 5 (субмодуль):
  - Все параметры GEOADDR анализируются.
  - Выводится аппаратный идентификатор субмодуля.

Параметр AREA типа системных данных GEOADDR не анализируется.

Таблица 9-226 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
GEOADDR	IN/OUT or IN ?	Variant	Указатель на структуру типа системных данных GEOADDR. Тип системных данных GEOADDR содержит информацию слота, из которой определяется аппаратный ID. Дополнительную информацию см. в "Системный тип данных GEOADDR (Страница 553)".
RET_VAL	OUT or RETURN ?	Int	Вывод информации об ошибках.
LADDR	OUT	HW_ANY	Аппаратный идентификатор узла или модуля. Номер присваивается автоматически и сохраняется в свойствах конфигурации оборудования.

Дополнительную информацию о допустимых типах данных можно найти в разделе "Обзор поддерживаемых типов данных" в системе интерактивной помощи STEP 7.

Таблица 9-227 Коды условий

RET_VAL* (W#16#...)	Объяснение
0	Ошибки отсутствуют.
8091	Недействительное значение в GEOADDR для HWTYPE.
8094	Недействительное значение в GEOADDR для IOSYSTEM.
8095	Недействительное значение в GEOADDR для STATION.
8096	Недействительное значение в GEOADDR для SLOT.
8097	Недействительное значение в GEOADDR для SUBSLOT.
* Коды ошибок могут отображаться в редакторе текстов программ в виде целых или шестнадцатеричных значений.	

9.11.2 LOG2GEO (определение слота по аппаратному идентификатору)

Инструкция LOG2GEO используется для определения географического адреса (слота модуля) из логического адреса аппаратного идентификатора.

Таблица 9-228 Инструкция LOG2GEO

LAD/FBD	SCL	Описание

	`ret_val := LOG2GEO(`
	`  laddr:=_word_in_,`
	
	`GEOADDR:=_variant_in_out_);`
		С помощью инструкции LOG2GEO можно определить слот модуля, относящийся к аппаратному идентификатору.

Инструкция LOG2GEO определяет географический адрес логического адреса на основе аппаратного идентификатора:

- С помощью параметра LADDR на основе аппаратного идентификатора выбирается логический адрес.
- Параметр GEOADDR содержит географический адрес указанного на входе LADDR логического адреса.

Примечание

Если тип HW не поддерживает компонент, то возвращается номер места субслота для модуля 0.

Если вход LADDR не адресует HW-объект, то выводится ошибка.

Таблица 9-229 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
LADDR	IN	HW_ANY	Аппаратный идентификатор система ввода/вывода или модуля. Номер присваивается автоматически и сохраняется в свойствах CPU или интерфейса конфигурации оборудования.
RET_VAL	OUT	Int	Код ошибки инструкции
GEOADDR	IN_OUT	Variant	Указатель на тип системных данных GEOADDR. Тип системных данных GEOADDR содержит информацию слота. Дополнительную информацию см. в "Системный тип данных GEOADDR (Страница 553)".

Дополнительную информацию о допустимых типах данных можно найти в разделе "Обзор поддерживаемых типов данных" в системе интерактивной помощи STEP 7.

Таблица 9-230 Коды условий

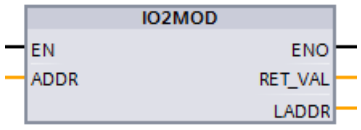
RET_VAL (W#16#...)	Описание
0000	Ошибки отсутствуют
8090	Адрес, указанный в параметре LADDR, является недействительным.
* Коды ошибок могут отображаться в редакторе текстов программ в виде целых или шестнадцатеричных значений.	

9.11.3 IO2MOD (определение аппаратного идентификатора по адресу ввода/вывода)

С помощью инструкции IO2MOD аппаратный идентификатор модуля определяется по адресу ввода/вывода субмодуля.

Таблица 9-231 Инструкция IO2MOD

LAD/FBD	SCL	Описание

	`ret_val := IO2MOD(`
	`  ADDR:=_word_in_,`
	`  LADDR:=_word_out_);`
		С помощью инструкции IO2MOD можно определить слот модуля, относящийся к аппаратному идентификатору.

Инструкция IO2MOD определяет аппаратный идентификатор модуля по адресу ввода/вывода (I, Q, PI, PQ) субмодуля.

Ввести адрес ввода/вывода в параметр ADDR. Если для этого параметра используется несколько адресов ввода/вывода, то для определения аппаратного идентификатора анализируется только первый адрес. Если первый адрес задан корректно, то длина для спецификации адреса в параметре ADDR не имеет значения. Если используемая адресная область включает в себя несколько модулей или неиспользуемых адресов, то также может быть определен аппаратный идентификатор первого модуля.

Если в параметре ADDR не указан адрес ввода/вывода (суб)модуля, то в параметре RET_VAL выводится код ошибки "8090".

Примечание

Ввод адреса ввода/вывода в SCL

Нельзя использовать для программирования, I/O идентификатор доступа "%QWx:P" в SCL. В этом случае следует использовать символьное имя переменной (тега) или абсолютный адрес в образе процесса.

Таблица 9-232 Типы данных для параметров

Параметр	Объявление	Тип данных	Область памяти	Описание
ADDR	IN or IN/OUT ?	Variant	I, Q, M, D, L	Адрес ввода/вывода (I, Q, PI, PQ) в субмодуле. Проследить, чтобы для параметра ADDR не использовался Slice-доступ. В этом случае в параметре LADDR выводились бы неправильные значения.
RET_VAL	OUT or RETURN ?	Int	I, Q, M, D, L	Код ошибки инструкции.
LADDR	OUT	HW_IO	I, Q, M, D, L	Определенный аппаратный идентификатор (логический адрес) субмодуля ввода/вывода.

Дополнительную информацию о допустимых типах данных можно найти в разделе "Обзор поддерживаемых типов данных" в системе интерактивной помощи STEP 7.

Таблица 9-233 Коды условий

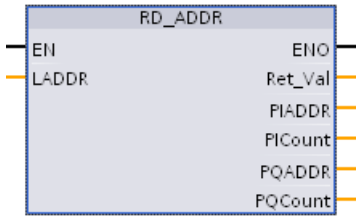
RET_VAL* (W#16#...)	Объяснение
0	Ошибки отсутствуют.
8090	Указанный в параметре ADDR адрес ввода/вывода не используется ни одним аппаратным компонентом.
* Коды ошибок могут отображаться в редакторе текстов программ в виде целых или шестнадцатеричных значений.	

9.11.4 RD_ADDR (определение адреса ввода/вывода по аппаратному идентификатору)

С помощью инструкции RD_ADDR запрашивается адрес ввода/вывода субмодуля.

Таблица 9-234 Инструкция RD_ADDR

LAD/FBD	SCL	Описание

	`ret_val := RD_ADDR(`
	`laddr:=_word_in_,`
	`PIADDR=>_udint_out_,`
	`PICount=>_uint_out_,`
	`PQADDR=>_udint_out_,`
	`PQCount=>_uint_out_,);`
		С помощью инструкции RD_ADDR запрашивается адрес ввода/вывода субмодуля.

Инструкция RD_ADDR на основании аппаратного идентификатора субмодуля определяет длину и начальный адрес входов или выходов.

- С помощью параметра LADDR на основе аппаратного идентификатора выбирается модуль входов или выходов.
- В зависимости от того, идет ли речь о модуле входов или модуле выходов модуле, используются следующие выходные параметры:
  - В модуле входов определенные значения выводятся в параметрах PIADDR и PICOUNT.
  - В модуле выходов определенные значения выводятся в параметрах PQADDR и PQCOUNT.
- Параметры PIADDR и PQADDR содержат начальные адреса ввода/вывода модуля соответственно.
- Параметры PICOUNT иPQCOUNT содержат соответствующее количество байт входов или выходов (1 байт для 8 входов/выходов, 2 байта для 16 входов/выходов).

Таблица 9-235 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
LADDR	IN	HW_IO	Аппаратный идентификатор (суб)модуля
RET_VAL	OUT	Int	Код ошибки инструкции
PIADDR	OUT	UDInt	Начальный адрес модуля входов
PICOUNT	OUT	UInt	Количество байт входов
PQADDR	OUT	UDInt	Начальный адрес модуля выходов
PQCOUNT	OUT	UInt	Количество байт выходов

Дополнительную информацию о допустимых типах данных можно найти в разделе "Обзор поддерживаемых типов данных" в системе интерактивной помощи STEP 7.

Таблица 9-236 Коды условий

RET_VAL (W#16#...)	Описание
0000	Ошибки отсутствуют
8090	Недействительный аппаратный идентификатор модуля в параметре LADDR.
* Коды ошибок могут отображаться в редакторе текстов программ в виде целых или шестнадцатеричных значений.	

9.11.5 Тип системных данных GEOADDR

Географический адрес

Тип системных данных GEOADDR содержит географический адрес модуля (или информацию слота).

- Географический адрес для PROFINET IO:
  Географический адрес для PROFINET IO состоит из идентификатора системы PROFINET IO, номера устройства, номера слота и субмодуля (если субмодуль используется).
- Географический адрес для PROFIBUS DP:
  Географический адрес для PROFIBUS DP состоит из идентификатора система ведущих устройств DP, номера станции и номера слота.

Информацию слота модулей можно найти в конфигурации оборудования каждого модуля.

Структура типа системных данных GEOADDR.

Структура GEOADDR создается автоматически, когда в блок данных в качестве типа данных вводится "GEOADDR".

Имя параметра	Тип данных	Описание
GEOADDR	STRUCT	
HWTYPE	UINT	Тип оборудования: • 1: IO-система (PROFINET/PROFIBUS) • 2: IO-устройство/ведомое устройство DP • 3: Монтажная стойка • 4: Модуль • 5: Субмодуль Если тип оборудования не поддерживается инструкцией, в параметре HWTYPE выводится значение "0".
AREA	UINT	Идентификатор области: • 0 = CPU • 1 = PROFINET IO • 2 = PROFIBUS DP • 3 = AS-i
IOSYSTEM	UINT	PROFINET IO система (0 = центральный модуль в стойке)
STATION	UINT	• Номер монтажной стойки, если идентификатор области AREA = 0 (центральный модуль). • Номер станции, если идентификатор области AREA > 0.
SLOT	UINT	Номер слота
SUBSLOT	UINT	Номер субмодуля Этот параметр имеет значение "0", если субмодуль отсутствует или не может быть вставлен.

9.12 Общие коды ошибок для расширенных инструкций

Таблица 9-237 Общие коды условий для расширенных инструкций

Код условия (W#16#....)[1]	Описание
8x22[2]	Диапазон для входа слишком маленький
8x23	Диапазон для выхода слишком маленький
8x24	Недопустимая область ввода
8x25	Недопустимая область вывода
8x28	Недопустимое назначение входного бита
8x29	Недопустимое назначение выходного бита

Код условия (W#16#....)[1]	**Описание**
8x30	Область вывода это DB с защитой от записи.
8x3A	Блок данных отсутствует.

[1] Если одна из этих ошибок возникает при выполнении блока кода, то CPU остается в рабочем режиме RUN (по умолчанию) или переходит в режим STOP при соответствующем конфигурировании. Как опция с помощью инструкций GetError или GetErorID в этом блоке кода можно запустить локальную обработку ошибок (CPU остается в режиме RUN) с программируемой реакцией на ошибки.

[2] "x" - это номер параметра с ошибкой. Нумерация параметров начинается с 1.

Технологические инструкции 10

10.1 Счет (высокоскоростные счетчики)

Основные описываемые в "Счетчиках" (Страница 236) счетные инструкции, подсчитывают только такие события, которые возникают медленнее, чем цикл S7-1200 CPU. Высокоскоростной счетчик (High-Speed Counter, HSC) позволяет считать импульсы, возникающие быстрее цикла PLC. Кроме этого, HSC может быть сконфигурирован таким образом, что он будет считать частоту и период возникающих импульсов, или HSC может быть настроен для чтения сигнала датчика двигателя со стороны управления перемещением.

Для использования функций HSC, сначала HSC должен быть активирован и настроен в конфигурации устройства на вкладке "Свойства" модуля CPU. Процесс конфигурирования HSC описаны в "Конфигурирование высокоскоростного счетчика" (Страница 573).

После загрузки конфигурации оборудования, HSC может подсчитывать импульсы или мерить частоту без вызова соответствующих инструкций. Если HSC находится в режиме счета или периода, то подсчитанное значение фиксируется автоматически в каждом цикле и обновляется в образе процесса (область памяти I). Если HSC находится в режиме частоты, то значение в образе процесса это частота в Гц.

Наряду со счетом и измерением, HSC также может генерировать события аппаратных прерываний, работать в зависимости от состояния физических входов и создавать выходной импульс согласно указанному событию счетчика (только для CPU от V4.2). С помощью технологической инструкции CTRL_HSC_EXT программа пользователя может управлять HSC через программу. Если выполняется CTRL_HSC_EXT, то инструкция обновляет параметры HSC и возвращает последние значения. Инструкция CTRL_HSC_EXT может использоваться в режимах счета, периода и частоты.

Примечание

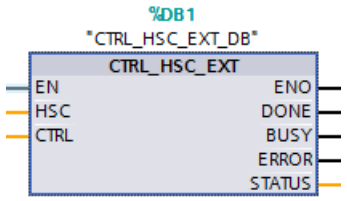
Инструкция CTRL_HSC_EXT заменяет прежнюю инструкцию CTRL_HSC для проектов для CPU от V4.2. Инструкция CTRL_HSC_EXT предлагает весь объем функций инструкции CTRL_HSC, а также дополнительные функции. Прежняя инструкция CTRL_HSC только по причине совместимости доступна для старых проектов S7-1200 и не должна использоваться в новых проектах.

10.1.1 Инструкции CTRL_HSC_EXT (управление высокоскоростным счетчиком)

10.1.1.1 Обзор инструкции

Таблица 10-1 Инструкция CTRL_HSC_EXT

LAD/FBD	SCL	Описание

	`"CTRL_HSC_1_DB" (`
	`hsc:=_hw_hsc_in_,`
	`done:=_done_out_,`
	`busy:=_busy_ out_,`
	`error:=_error_out_,`
	`status:=_status_out_,`
	`ctrl:=_variant_in_);`
		Для сохранения данных счетчика, каждая инструкция CTRL_HSC_EXT (управление высокоскоростным счетчиком (расширенное)) использует сохраненную в пользовательском глобальном DB, определенную системой структуру данных. Типы данных HSC_Count, HSC_Period и HSC_Frequency назначаются как входные параметры инструкции CTRL_HSC_EXT.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "CTRL_HSC_1_DB" - это имя DB экземпляра.

Таблица 10-2 Типы данных для параметров

Параметр	Объявление	Тип данных	Описание
HSC	IN	HW_HSC	Идентификатор HSC
CTRL	IN_OUT	Variant	Вход SFB и возвращаемые данные. Примечание: Дополнительную информацию можно найти здесь: "Типы системных данных для инструкции CTRL_HSC_EXT (SDT) (Страница 562)".
DONE	OUT	Bool	1 = Показывает, что SFB завершен. Всегда 1, т.е. SFB является синхронным.
BUSY	OUT	Bool	Всегда 0, функция никогда не занята.
ERROR	OUT	Bool	1 = Указание на ошибку.
STATUS	OUT	Word	Условие выполнения Примечание: См. следующую таблицу "Условие выполнения" с дополнительной информацией.

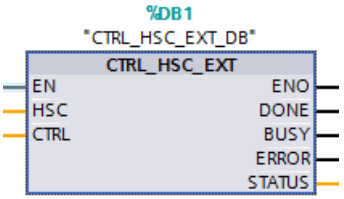
Таблица 10-3 Условие выполнения

STATUS (W#16#)	Описание
0	Ошибки отсутствуют
80A1	Идентификатор HSC не адресует HSC
80B1	Недействительное значение в NewDirection.
80B4	Недействительное значение в NewPeriod.
80B5	Недействительное значение в NewOpModeBehavior.
80B6	Недействительное значение в NewLimitBehavior.
80D0	SFB 124 недоступен

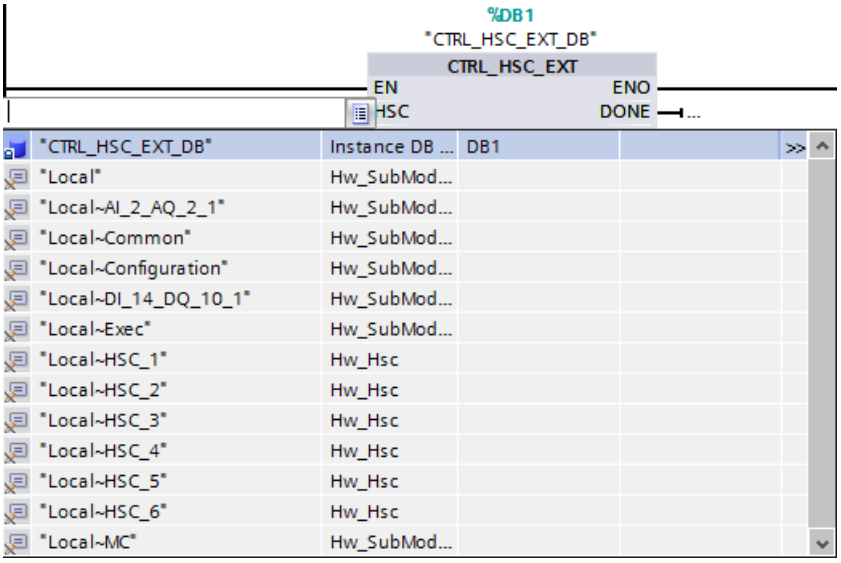
10.1.1.2 Пример

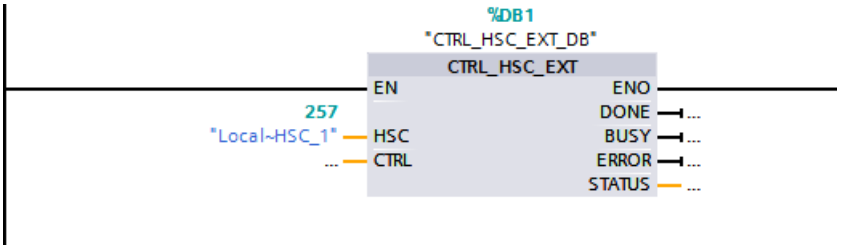
Для использования инструкции CTRL_HSC_EXT выполнить следующие действия:

1. Поместить инструкцию CTRL_HSC_EXT в сегмент LAD, который также создает блок данных экземпляра: "CTRL_HSC_EXT_DB":



2. Добавить аппаратный идентификатор HSC, который находится в свойствах HSC, на соединитель "HSC" инструкции LAD. Также можно выбрать один из шести объектов "Hw_Hsc" из выпадающего меню для этого входного соединителя. Имя переменной (тега) по умолчанию для HSC1 это "Local~HSC_1":



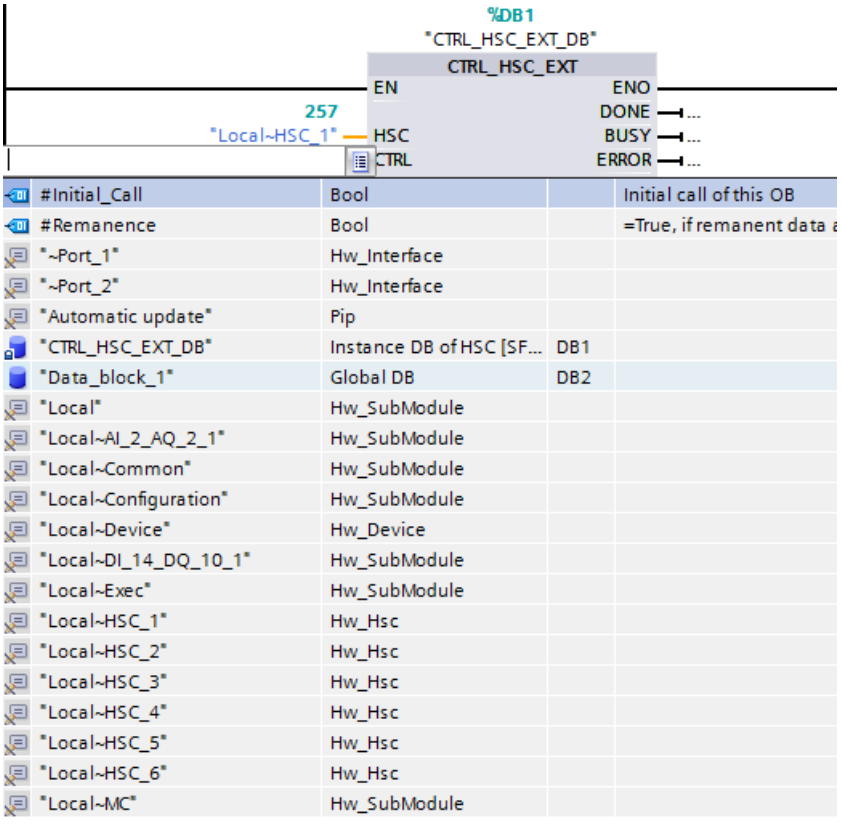


3. Создать глобальный блок данных с именем "Data_block_1" (также можно использовать существующий глобальный блок данных):
   - Найти в "Data_block_1" пустую строку и добавить переменную с имеем "MyHSC".
   - Ввести в столбце "Тип данных" один из следующих типов системных данных (SDT). Выбрать SDT, соответствующий сконфигурированному методу счета HSC. Дополнительную информацию о SDT для HSC можно найти ниже в данном разделе. В выпадающем списке эти типы отсутствуют. Поэтому обязательно проследить, чтобы имя SDT было бы введено точно так, как показано: HSC_Count, HSC_Period или HSC_Frequency
   - После ввода типа данных можно развернуть переменную "MyHSC", чтобы просмотреть все поля, содержащиеся в структуре данных. Здесь можно найти тип данных каждого из полей и изменить стартовые значения по умолчанию:

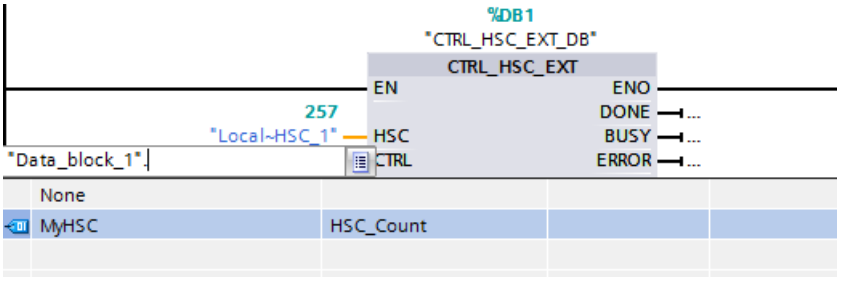
Data_block_1

	Name	Data type	Start value
1	Static		
2	MyHSC	HSC_Count	
3	CurrentCount	DInt	0
4	CapturedCount	DInt	0
5	SyncActive	Bool	false
6	DirChange	Bool	false
7	CmpResult_1	Bool	false
8	CmpResult_2	Bool	false
9	OverflowNeg	Bool	false
10	OverflowPos	Bool	false
11	EnHSC	Bool	false
12	EnCapture	Bool	false
13	EnSync	Bool	false
14	EnDir	Bool	false
15	EnCV	Bool	false
16	EnSV	Bool	false
17	EnReference1	Bool	false
18	EnReference2	Bool	false
19	EnUpperLmt	Bool	false
20	EnLowerLmt	Bool	false
21	EnOpMode	Bool	false
22	EnLmtBehavior	Bool	false
23	EnSyncBehavior	Bool	false
24	NewDirection	Int	0
25	NewOpModeBeha...	Int	0
26	NewLimitBehavior	Int	0
27	NewSyncBehavior	Int	0
28	NewCurrentCount	DInt	0
29	NewStartValue	DInt	0
30	NewReference1	DInt	0
31	NewReference2	DInt	0
32	NewUpperLimit	DInt	0
33	New_Lower_Limit	DInt	0

4. Назначить переменную "'Data_block_1'. MyHSC" входному соединителю CTRL инструкции CTRL_HSC_EXT:
   - Выбрать "Data_Block_1".



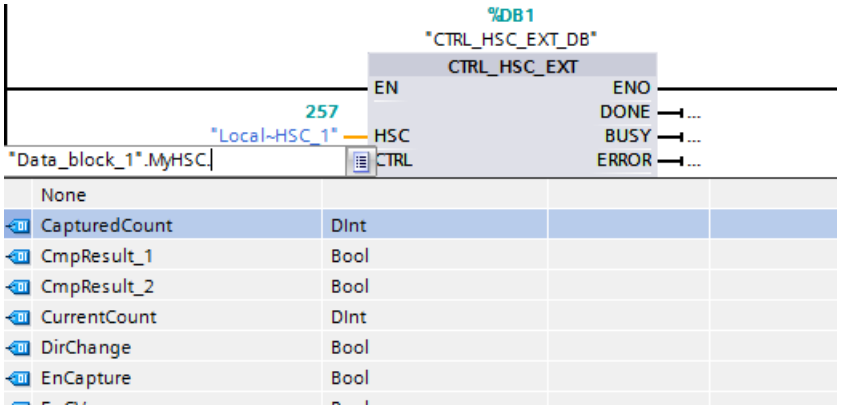
– Выбрать "MyHSC".



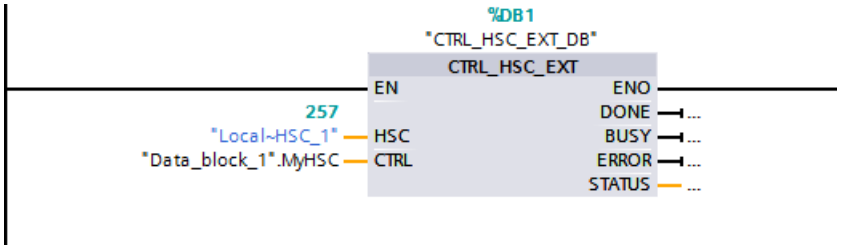
– Удалить точку ("."), следующую за "'Data_Block_1'.MyHSC". Кликнуть мышью вне поля или нажать клавишу ESC и после клавишу ввода.

Примечание

После удаления точки ("."), следующей за "'Data_Block_1'.MyHSC", не нажимать только клавишу ввода. Иначе точка (".") снова будет вставлена в поле.



– Полный ввод для CTRL показан ниже.



После конфигурирования HSC в PLC, можно выполнить инструкцию CTRL_HSC_EXT. При ошибке ENO устанавливается на 0 и выход STATUS возвращает код условия.

См. также

Типы системных данных (SDT) инструкции CTRL_HSC_EXT (Страница 562)

10.1.1.3 Типы системных данных (SDT) инструкции CTRL_HSC_EXT

Следующие типы системных данных (SDT) используются только для соединителя CTRL инструкции CTRL_HSC_EXT. Для их использования следует создать блок данных пользователя и добавить в него объект с типом данных SDT, соответствующий сконфигурированному режиму работы (режиму счета) счетчика HSC. В STEP 7 эти типы данных не отображаются в выпадающем меню. Ввести имя SDT точно как показано.

Входы SDT счетчика HSC обозначаются префиксом "En" или "New". Входы с префиксом "En" либо активируют функцию HSC, либо обновляют соответствующий параметр. Префикс "New" идентифицирует значения параметров операции обновления данных. Для того, чтобы новое значение могло быть активировано, соответствующий бит "En" должен быть TRUE и значение "New" должно быть действительным. При выполнении инструкции CTRL_HSC_EXT, программа применяет изменения на входах и обновляет выходы с соответствующими значениями SDT.

SDT: HSC_Count

Тип данных "HSC_Count" соответствует HSC, сконфигурированному для режима работы "счет". Счетный режим работы предлагает следующие функции:

- Доступ к текущему значению подсчитанных импульсов
- Фиксация текущего числа отсчитанных импульсов при событии на входе
- Сброс текущего числа отсчитанных импульсов на стартовое значение при событии на входе
- Доступ к битам состояния, сигнализирующим наступление определенных событий HSC
- Деактивация HSC через программный или аппаратный вход
- Изменение направления счета через программный или аппаратный вход
- Изменение текущего числа отсчитанных импульсов
- Изменение стартового значения (используется, если CPU переходит в RUN или при запуске функции синхронизации)
- Изменение двух независимых опорных значений (или предустановленных значений) для сравнений
- Изменение верхней или нижней границ счета
- Изменение принципа работы HSC в случае, когда число отсчитанных импульсов достигает этих предельных значений
- Генерирование события аппаратного прерывания при достижении текущим числом отсчитанных импульсов опорного значения (предустановленного значения)
- Генерирование события аппаратного прерывания при активации входа синхронизации (R-входа)
- Генерирование события аппаратного прерывания при изменении направления счета на базе внешнего входа
- Генерация одиночного выходного импульса при указанном событии счета

Если при выполненной инструкции CTRL_HSC_EXT возникает событие, то инструкция устанавливает бит состояния. При следующем выполнении инструкции, она удаляет бит состояния, если событие не возникает снова до выполнения инструкции.

Таблица 10-4 Структура HSC_Count

Элемент структуры	Объявление	Тип данных	Описание
CurrentCount	OUT	DInt	Возвращает текущее значение счета HSC
CapturedCount	OUT	DInt	Возвращает зафиксированное на указанном входном событии значение счета
SyncActive	OUT	Bool	Бит состояния: Вход синхронизации был активирован
DirChange	OUT	Bool	Бит состояния: Направление счета было изменено
CmpResult1	OUT	Bool	Бит состояния: Имеет место событие CurrentCount, идентичное Reference1
CmpResult2	OUT	Bool	Бит состояния: Имеет место событие CurrentCount, идентичное Reference2

Элемент структуры	Объявление	Тип данных	Описание
OverflowNeg	OUT	Bool	Бит состояния: CurrentCount достиг LowerLimit
OverflowPos	OUT	Bool	Бит состояния: CurrentCount достиг UpperLimit
EnHSC	IN	Bool	При true разрешает HSC считать импульсы; если false, то подсчет отключается
EnCapture	IN	Bool	При true активирует вход Capture; если false, то вход Capture не действует
EnSync	IN	Bool	При true активирует вход Sync; если false, то вход Sync не действует
EnDir	IN	Bool	Позволяет активировать значение NewDirection
EnCV	IN	Bool	Позволяет активировать значение NewCurrentCount
EnSV	IN	Bool	Позволяет активировать значение NewStartValue
EnReference1	IN	Bool	Позволяет активировать значение NewReference1
EnReference2	IN	Bool	Позволяет активировать значение NewReference2
EnUpperLmt	IN	Bool	Позволяет активировать значение NewUpperLimit
EnLowerLmt	IN	Bool	Позволяет активировать значение New_Lower_Limit
EnOpMode	IN	Bool	Позволяет активировать значение NewOpModeBehavior
EnLmtBehavior	IN	Bool	Позволяет активировать значение NewLimitBehavior
EnSyncBehavior	IN	Bool	Это значение не используется.
NewDirection	IN	Int	Направление счета: 1 = прямой счет; -1 = обратный счет; все остальные значения зарезервированы
NewOpModeBehavior	IN	Int	Принцип действия HSC при переполнении: 1 = HSC останавливает процесс счета (для возобновления процесса счета потребуется выключить и снова включить HSC); 2 = HSC продолжает работу; все другие значения зарезервированы
NewLimitBehavior	IN	Int	Результат значения CurrentCount при переполнении: 1 = установить CurrentCount на противоположное предельное значение; 2 = установить CurrentCount на StartValue; все другие значения зарезервированы
NewSyncBehavior	IN	Int	Это значение не используется.
NewCurrentCount	IN	DInt	Значение CurrentCount
NewStartValue	IN	DInt	StartValue: начальное значение HSC
NewReference1	IN	DInt	Значение Reference1
NewReference2	IN	DInt	Значение Reference2
NewUpperLimit	IN	DInt	Верхнее предельное значение счета
New_Lower_Limit	IN	DInt	Нижнее предельное значение счета

SDT: HSC_Period

Тип данных "HSC_Period" соответствует HSC, сконфигурированному для режима работы "период". Инструкция CTRL_HSC_EXT предлагает обращение из программы к числу входных импульсов за указанный интервал измерения. Эта инструкция позволяет рассчитать период времени между входными импульсами с точным наносекундным разрешением.

Таблица 10-5 Структура HSC_Period

Элемент структуры	Объявление	Тип данных	Описание
ElapsedTime	OUT	UDInt	См. описание ниже.
EdgeCount	OUT	UDInt	См. описание ниже.
EnHSC	IN	Bool	Если true, активирует HSC для измерения периода; деактивирует измерение периода при false.
EnPeriod	IN	Bool	Позволяет активировать значение NewPeriod.
NewPeriod	IN	Int	Указывает интервал измерения в миллисекундах. Допускаются только значения 10, 100 или 1000 мс.

ElapsedTime возвращает время в наносекундах между последними событиями счета последовательных интервалов измерения. Если в течение интервала измерения не произошло никаких событий счета, ElapsedTime возвращает совокупное время с момента последнего события счета. ElapsedTime имеет диапазон от "0" до 4.294.967.280 наносекунд (от 0x0000 0000 до 0xFFFF FFF0). Возвращаемое значение 4.294.967.295 (0xFFFF FFFF) сигнализирует переполнение периода. Переполнение указывает на то, что время между фронтами импульсов превышает 4,295 секунды, и период нельзя рассчитать с помощью этой инструкции. Значения от 0xFFFF FFF1 до 0xFFFF FFFE зарезервированы.

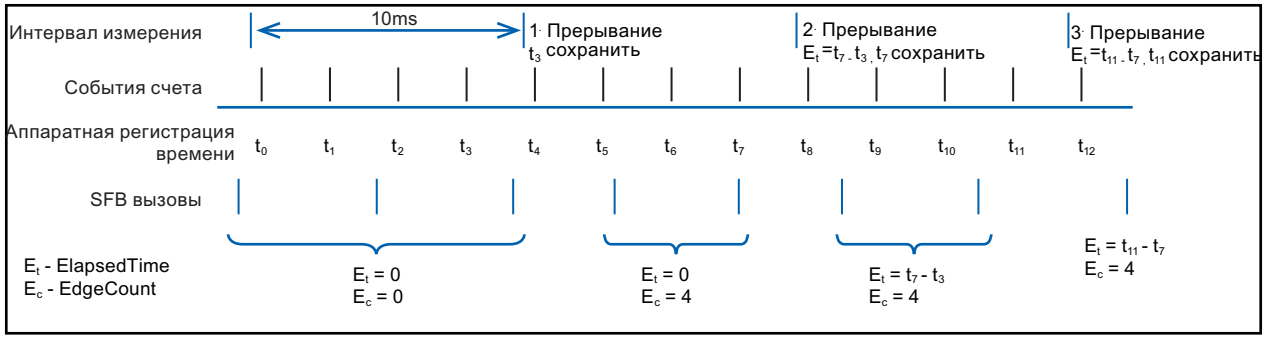
EdgeCount возвращает количество событий подсчета, полученных в течение интервала измерения. Период может быть рассчитан только в том случае, если значение EdgeCount больше нуля. Если ElapsedTime имеет значение 0 (входные импульсы не получены) или 0xFFFF FFFF (переполнение периода), то значение EdgeCount является недействительным.

Когда EdgeCount является действительным, следует использовать следующую формулу для расчета периода в наносекундах: Период = ElapsedTime/EdgeCount

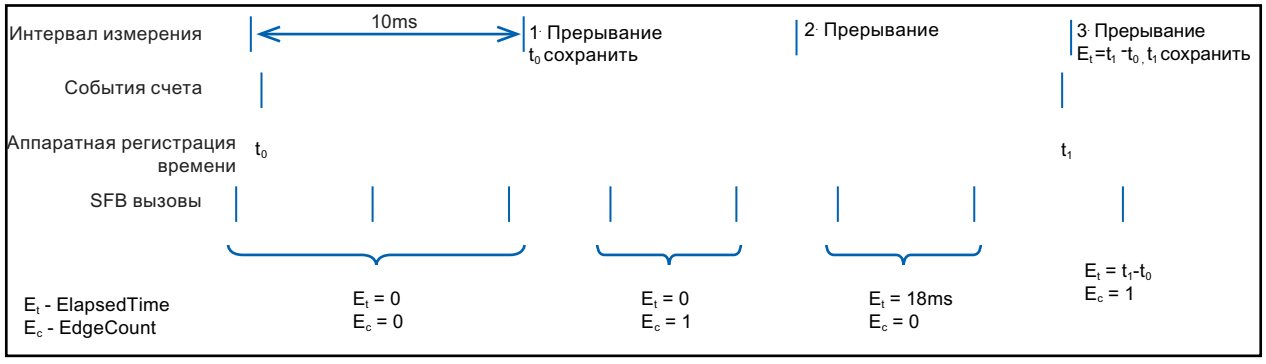
Рассчитанное значение периода является средним из периодов времени всех импульсов, которые происходят в течение интервала измерения. Если период входящего импульса больше интервала измерения (10, 100 или 1000 мс), то для расчета периода требуется несколько интервалов измерения.

Следующие примеры показывают, как инструкция выполняет измерения периода:

Пример 1: Несколько событий счета за один интервал измерения



Пример 2: Ноль и одно событие счета за несколько интервалов измерения



Правила:
1. Если $E1_t$ = 0, значит период времени неправильный
2. В остальных случаях, период времени = E_t / E_c

SDT: HSC_Frequency

Тип данных "HSC_Frequency" соответствует HSC, сконфигурированному для режима "частота". Инструкция CTRL_HSC_EXT обеспечивает программный доступ к частоте входных импульсов, измеренных за определенный период времени.

Инструкция CTRL_HSC_EXT в частотном режиме работы предлагает следующие возможности:

Таблица 10-6 Структура HSC_Frequency

Элемент структуры	Объявление	Тип данных	Описание
Частота	OUT	DInt	Возвращает частоту в Гц, измеренную в течение интервала измерения. Когда HSC выполняет обратный счет, инструкция возвращает отрицательную частоту.
EnHSC	IN	Bool	Если true, активирует HSC для измерения частоты; деактивирует измерение частоты при false.
EnPeriod	IN	Bool	Позволяет активировать значение NewPeriod.
NewPeriod	IN	Int	Указывает интервал измерения в миллисекундах. Допускаются только значения 10, 100 или 1000 мс.

Инструкция CTRL_HSC_EXT использует для измерения частоты такую же технику, что используется в режиме работы "период" для получения ElapsedTime и EdgeCount. Инструкция вычисляет частоту в виде целочисленного значения со знаком в Гц по формуле: Частота = EdgeCount/ElapsedTime

Если для частоты требуется значение с плавающей запятой, то можно использовать приведенную выше формулу для частоты, когда HSC находится в режиме периода. Обратить внимание, что в режиме периода значение ElapsedTime возвращается в наносекундах и может потребоваться его масштабирование.

10.1.2 Работа высокоскоростного счётчика

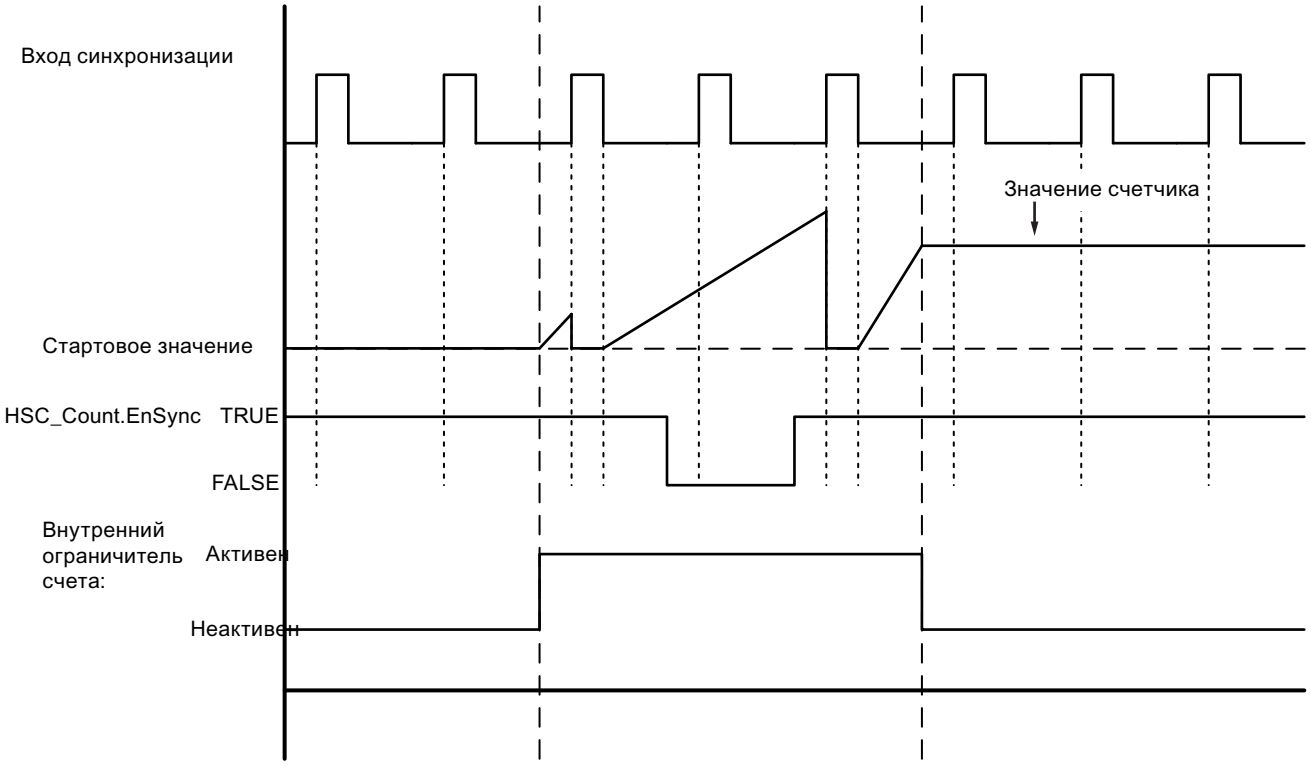
10.1.2.1 Функция синхронизации

С помощью функции синхронизации счетчик устанавливается на стартовое значение с помощью внешнего входного сигнала. Можно изменить стартовое значение, выполнив инструкцию CTRL_HSC_EXT. Это позволяет синхронизировать текущее значение счетчика с требуемым значением при появлении внешнего входного сигнала.

Синхронизация всегда происходит при появлении входного сигнала и не зависит от состояния внутреннего ограничителя счета. Необходимо установить бит "HSC_Count.EnSync" на значение true, чтобы включить функцию синхронизации.

Инструкция CTRL_HSC_EXT устанавливает бит состояния HSC_Count.SyncActive на true после завершения синхронизации. Инструкция CTRL_HSC_EXT устанавливает бит состояния HSC_Count.SyncActive на значение false, если синхронизация не произошла с момента последнего выполнения инструкции.

На рисунке ниже показан пример для синхронизации, когда входной сигнал настроен на уровень "активный высокий":



Примечание

Сконфигурированные входные фильтры задерживают управляющий сигнал цифрового входа.

Эта входная функция доступна, только если HSC настроен на счетный режим работы.

Информацию о конфигурировании функции синхронизации можно найти во входных функциях (Страница 581).

10.1.2.2 Функция ограничения счета

Многие приложения требуют, чтобы процессы счета запускались или останавливались в соответствии с другими событиями. В таких случаях отсчет запускается и останавливается с использованием функции внутреннего ограничителя счета. Каждый канал HSC имеет два ограничителя счета: программный ограничитель счета и аппаратный ограничитель счета. Состояние этих ограничителей счета определяет состояние внутреннего ограничителя счета (см. следующую таблицу).

Внутренний ограничитель счета включен, если программный ограничитель счета включен, а аппаратный ограничитель счета включен или не был сконфигурирован. Если внутренний ограничитель счета включен, начинается подсчет. Если внутренний

ограничитель счета выключен, все остальные счетные импульсы игнорируются и счет останавливается.

Таблица 10-7 Состояния функции ограничителя счета

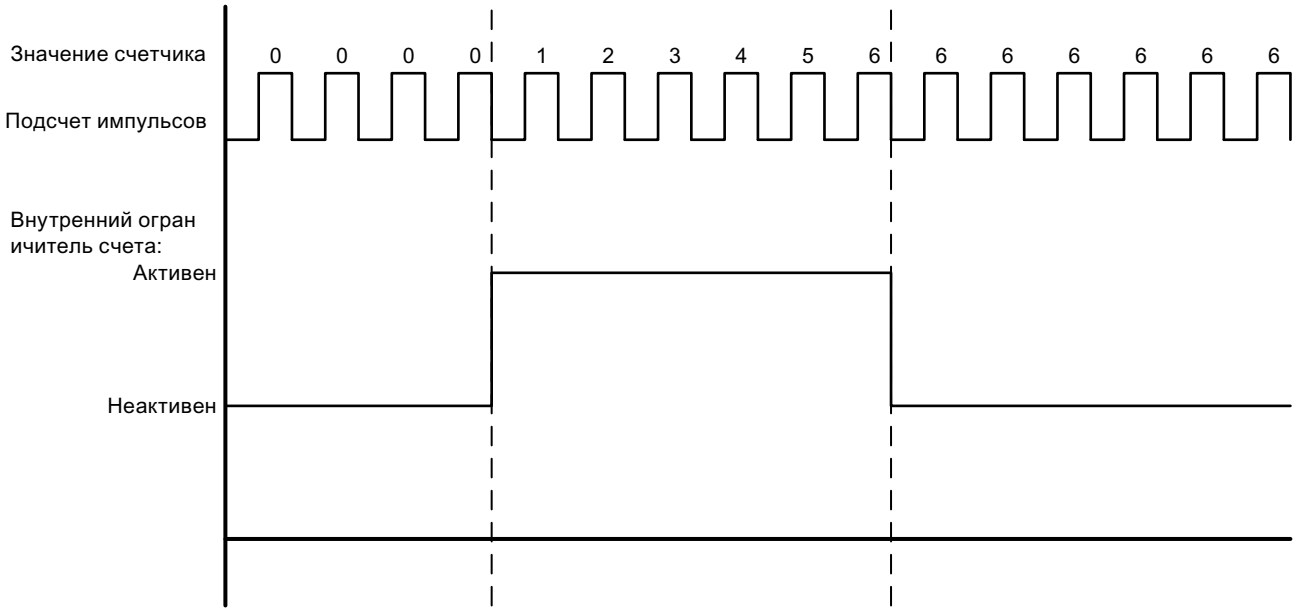
Аппаратный ограничитель счета	**Программный ограничитель счета**	**Внутренний ограничитель счета**
Включен/не сконфигурирован	Включен	Включен
Включен/не сконфигурирован	Выключен	Выключен
Выключен	Включен	Выключен
Выключен	Выключен	Выключен

Термин "включен" определен как активное состояние ограничителя счета. Соответственно термин "выключен" определен как неактивное состояние ограничителя счета.

Для управления программным ограничителем счета используется бит включения "HSC_Count.EnHSC" в SDT инструкции CTRL_HSC_EXT. Чтобы включить программный ограничитель счета, установить бит "HSC_Count.EnHSC" на значение true; чтобы выключить программный ограничитель счета, установить бит "HSC_Count.EnHSC" на значение false. Выполнить инструкцию CTRL_HSC_EXT, чтобы обновить состояние программного ограничителя счета.

Аппаратный ограничитель счета является опцией и может быть активирован или деактивирован в разделе свойств HSC. Для управления процессом счета с использованием только аппаратного ограничителя счета, программный ограничитель счета должен оставаться включенным. Если аппаратный ограничитель счета не конфигурируется, то он считается всегда включенным, а состояние внутреннего ограничителя счета совпадает с состоянием программного ограничителя счета.

На рисунке ниже показан пример включения и выключения для аппаратного ограничителя счета с цифровым входом. Цифровой вход настроен на активный высокий уровень:



Примечание

Сконфигурированные входные фильтры задерживают управляющий сигнал цифрового входа.

Функция аппаратного ограничителя счета доступна, только если HSC настроен на счетный режим работы. В режимах работы "период" и "частота" состояние внутреннего ограничителя счета идентично состоянию программного ограничителя счета.

В режиме работы "период" программный ограничитель счета управляется "HSC_Period.EnHSC".

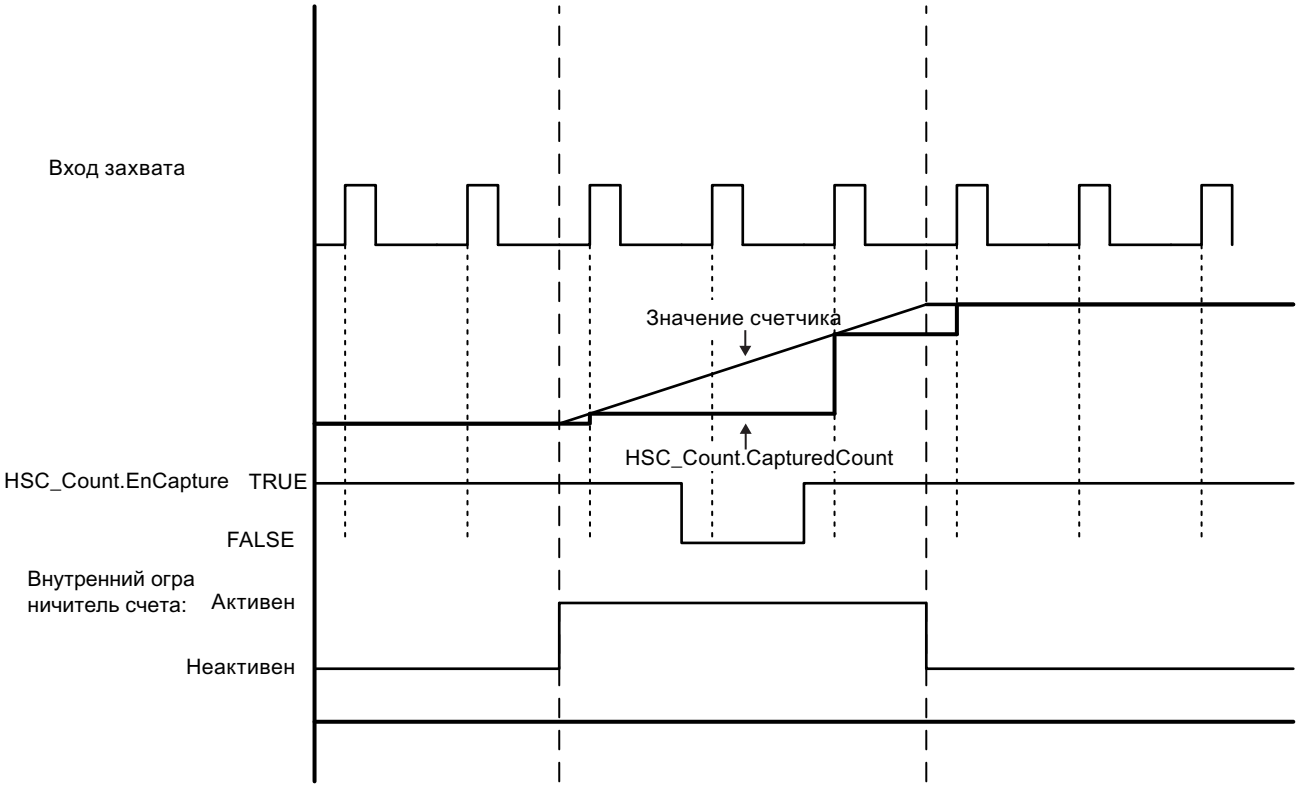
В режиме работы "частота" программный ограничитель счета управляется "HSC_Frequency.EnHSC".

Информацию о конфигурировании функции ограничителя счета можно найти во входных функциях (Страница 581).

10.1.2.3 Функция захвата

Функция захвата может использоваться для сохранения текущего значения счетчика с внешним опорным сигналом. При конфигурировании и активации с помощью бита "HSC_Count.EnCapture", функция захвата фиксируется текущее состояние счетчика при возникновении внешнего входного фронта. Функция захвата действует независимо от состояния внутреннего ограничителя счета. Если ограничитель счета выключен, то программа сохраняет значение счетчика без изменений. После выполнения инструкции CTRL_HSC_EXT программа сохраняет захваченное значение в "HSC_Count.CapturedCount".

На рисунке ниже показан пример функции захвата, сконфигурированной для захвата по переднему фронту. Вход захвата не запускает фиксацию текущего подсчитанного значения, если бит "HSC_Count.EnCapture" устанавливается на false через инструкцию CTRL_HSC_EXT.



Примечание

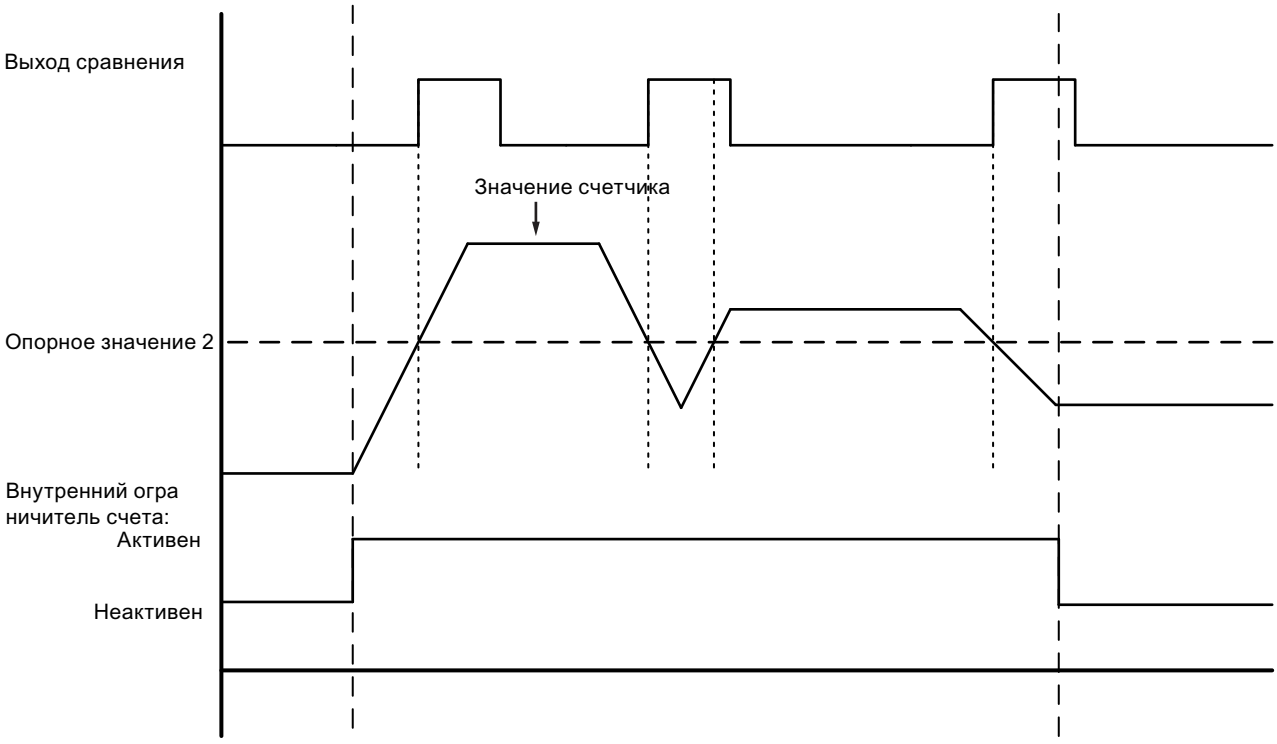
Сконфигурированные входные фильтры задерживают управляющий сигнал цифрового входа.

Эта входная функция может использоваться, только если HSC настроен на счетный режим работы.

Информацию о конфигурировании функции захвата можно найти во входных функциях (Страница 581).

10.1.2.4 Функция сравнения

При включении функция сравнения выходных сигналов генерирует один настраиваемый импульс, который возникает каждый раз, когда происходит сконфигурированное событие. К событиям относятся: Счетное значение, соответствующее одному из опорных значений или переполнению счетчика. Если импульс обрабатывается и событие происходит снова, импульс для этого события не создается.



Примечание

Эта выходная функция может использоваться, только если HSC настроен на счетный режим работы.

Информацию о конфигурировании функции сравнения можно найти во выходных функциях (Страница 582).

10.1.2.5 Применение

Типичное приложение использует HSC для контроля обратной связи от инкрементного датчика. Энкодер возвращает заданное количество счетных значений за оборот, которое можно использовать в качестве входа тактового генератора для HSC. Существует также импульс сброса, который возникает один раз за оборот и может использоваться в качестве входа синхронизации для HSC.

Для старта программа загружает первое опорное значение в HSC и устанавливает выходы в их начальные состояния. Выходы остаются в этом состоянии в течение периода времени, пока текущий счетчик меньше опорного значения. HSC создает прерывание,

когда текущее значение счетчика равно опорному значению, когда происходит событие синхронизации (сброс), а также, когда происходит изменение направления.

Если каждое значение счетчика равно опорному значению, происходит прерывающее событие. Программа загружает новое опорное значение в HSC и устанавливает выходы в их следующее состояния. Если происходит прерывающее событие синхронизации, программа устанавливает первое опорное значение и первые состояния выхода и повторяет цикл.

Поскольку прерывания происходят гораздо медленнее, чем скорость счета HSC, точное управление высокоскоростными операциями может быть реализовано с относительно небольшим влиянием на цикл CPU. Т.к. прерывания могут назначаться определенным прерывающим программам, каждая новая предустановка может быть загружена в отдельную программу обработки прерываний для простого управления состоянием. В качестве альтернативы все прерывающие события могут обрабатываться в одной прерывающей программе.

Функция ограничителя счета, запускаемая либо программой пользователя, либо внешним входным сигналом, может отключить подсчет импульсов энкодера. Можно игнорировать любое движение вала, отключив ограничитель счета. Это означает, что энкодер продолжает посылать импульсы на HSC, но значение счетчика удерживается на последнем значении до выключения ограничителя счета. После перехода ограничителя счета в активное состояние, счет продолжает с последнего значения перед отключением счетчика.

Функция захвата, если она активирована, позволяет фиксировать текущий счет при появлении внешнего события на входе. Процесс (например, процедура калибровки) может использовать эту функцию для определения количества импульсов между событиями.

Функция выхода сравнения, если она активирована, генерирует одиночный настраиваемый импульс каждый раз, когда текущее подсчитанное значение достигает одного из опорных значений или происходит переполнение (превышение пределов счета). Можно использовать этот импульс в качестве сигнала для запуска другого процесса всякий раз, когда происходит определенное событие HSC.

Направление счета управляется либо программой пользователя, либо внешним входным сигналом.

Чтобы запросить скорость вращения вала, можно настроить HSC на режим частоты. Эта функция возвращает целочисленное значение со знаком в Герцах. Поскольку сигнал сброса возникает один раз за оборот, измерение частоты сигнала сброса предоставляет быструю информацию о скорости вала в оборотах в секунду.

Если частота требуется в формате значения с плавающей запятой, следует настроить HSC для режима "период". Можно использовать значения ElapsedTime и EdgeCount, возвращаемые в режиме "период", для вычисления частоты.

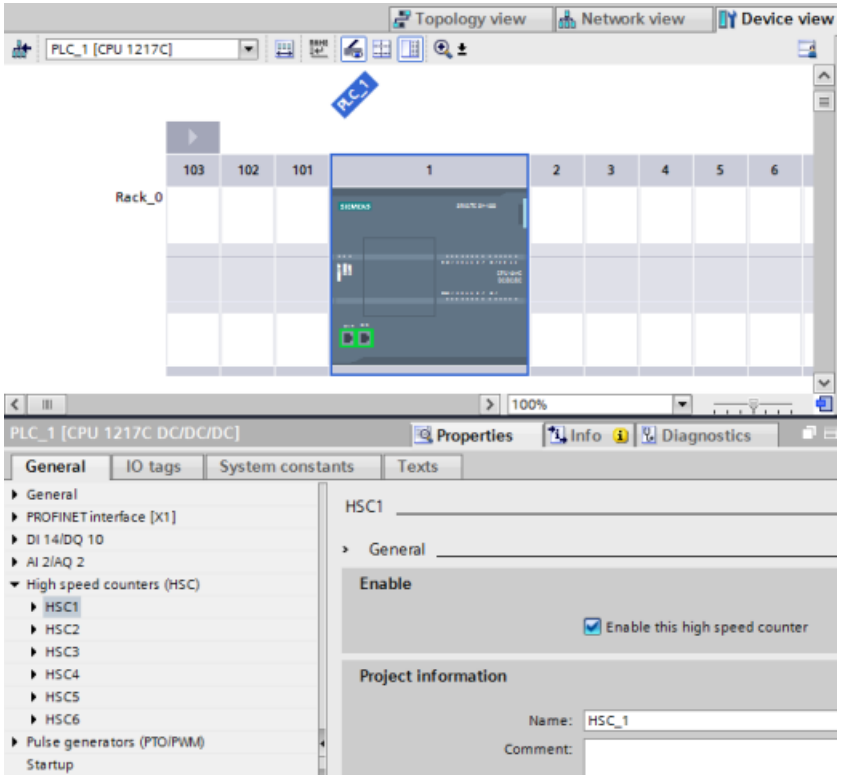
10.1.3 Конфигурирование высокоскоростного счетчика

Для настройки высокоскоростного счетчика (HSC) выполнить следующие действия:

- Выбрать конфигурацию устройства в дереве проекта.
- Выбрать CPU для настройки.

- Перейти в окне инспектора на вкладку "Свойства" (см. рисунок ниже).
- Выбрать HSC для активации в списке на вкладке "Общее" (см. рисунок ниже).

Можно настроить до шести высокоскоростных (HSC1 до HSC6). Активировать HSC путем выбора опции "Активировать этот высокоскоростной счётчик". После активации STEP 7 присваивает этому HSC стандартное однозначное имя. Это имя может быть изменено в поле редактирования "Имя:", при этом имя должно оставаться однозначным. Имена активированных HSC становятся переменными (тегами) с типом данных "HW_Hsc" в таблице переменных (тегов) "Системные константы" и могут использоваться в качестве параметра "HSC" инструкциями CTRL_HSC_EXT. Дополнительную информацию можно найти в Конфигурирование работы CPU (Страница 167).



После активации HSC, STEP 7 определяет однофазный счетчик как стандартную конфигурацию. Сразу после установки цифрового входного фильтра для входа тактового генератора HSC, программа может быть загружена в PLC, и CPU готов к счету. Как изменить конфигурацию HSC, описывается в следующем разделе "Типы счета".

В следующей таблице представлен обзор, какие входы и выходы доступны для каждой конфигурации:

Таблица 10-8 Типы счета для HSC

Тип	Вход 1	Вход 2	Вход 3	Вход 4	Вход 5	Выход 1	Функция
Однофазный счетчик с внутренним управлением направлением	Такт	-	-	-	-	-	Счет, направление или период
			Синхронизация	Ограничитель счета	Захват	Сравнение	Счет
Однофазный счетчик с внешним управлением направлением	Такт	Направление	-	-	-	-	Счет, направление или период
			Синхронизация	Ограничитель счета	Захват	Сравнение	Счет
Двухфазный счетчик	Счет в прямом направлении	Счет в обратном направлении	-	-	-	-	Счет, направление или период
			Синхронизация	Ограничитель счета	Захват	Сравнение	Счет
A/B-счетчик	Фаза A	Фаза B	-	-	-	-	Счет, направление или период
			Синхронизация[1]	Ограничитель счета	Захват	Сравнение	Счет
Квадратурный A/B-счетчик	Фаза A	Фаза B	-	-	-	-	Счет, направление или период
			Синхронизация[1]	Ограничитель счета	Захват	Сравнение	Счет

[1] Для энкодера: фаза Z, опорная точка

10.1.3.1 Типы счета

Существует четыре типа счета или режимов работы. При изменении режима работы, изменяются и доступные опции конфигурации для HSC:

- Счет: Подсчитывает количество импульсов и увеличивает или уменьшает значение счетчика в зависимости от состояния управления направлением. Внешний входы/выходы могут сбросить показания, отключить подсчет, инициировать захват текущего значения счетчика и генерировать одиночный импульс для определенного события. Выходными значениями являются текущее значение счетчика и значение счетчика в момент возникновения события захвата.
- Период: Подсчитывает количество входных импульсов за указанный период времени. Возвращает подсчитанное количество импульсов и длительность в наносекундах (нс). Значения фиксируются и рассчитываются в конце периода времени, указанного в периоде измерения частоты. Режим периода доступен для инструкции CTRL_HSC_EXT, но не для инструкции CTRL_HSC.

- Частота: Измеряет входные импульсы и длительность с вычислением частоты импульсов. Программа возвращает частоту в виде двойного целого числа со знаком в Герцах. Значение является отрицательным при обратной счете. Значения регистрируются и рассчитываются в конце периода времени, заданного периодом измерения частоты.
- Управление перемещением: Используется TO для управления перемещением и недоступен для инструкций HSC. Дополнительную информацию см. в: "Управление перемещением (Страница 656)".

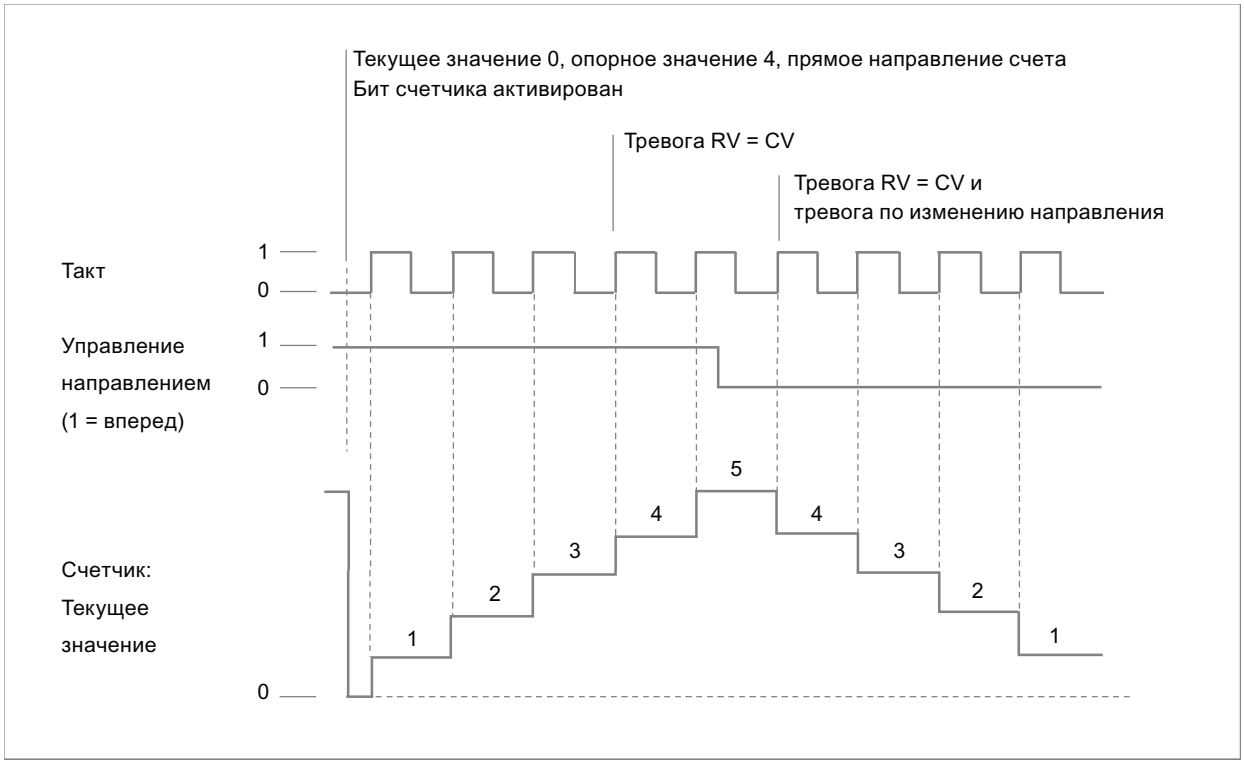
10.1.3.2 Рабочая фаза

Выбрать требуемую рабочую фазу HSC. Четыре рисунка ниже показывает, когда изменяется значение счетчика, когда происходит событие, что текущее значение (CV) равно опорному значению (RV), и когда происходит событие изменения направления.

Однофазный счетчик

Однофазный счетчик (недоступен для управления перемещением) считает импульсы:

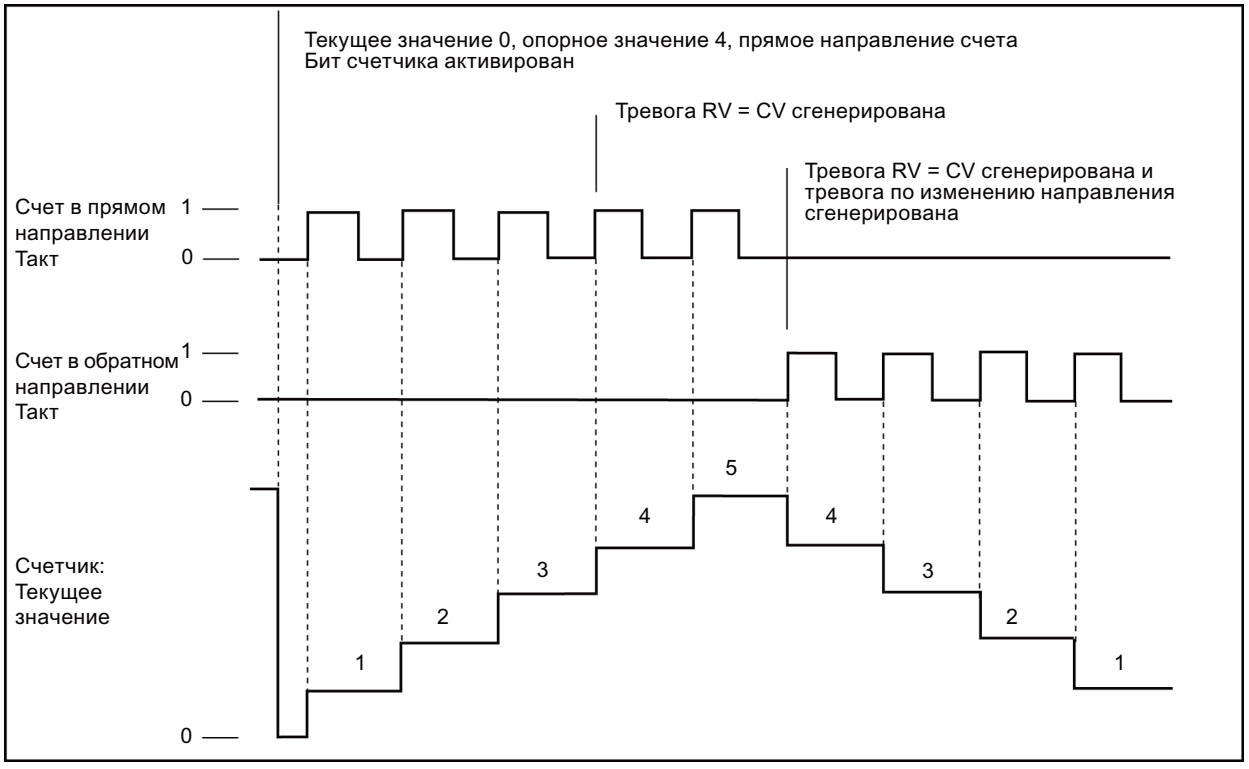
- Программа пользователя (внутреннее управление направлением):
  - 1 - вперед
  - - 1 - назад
- Аппаратный вход (внешнее управление направлением):
  - Высокий уровень - это счет в прямом направлении (вперед).
  - Низкий уровень - это счет в обратном направлении (назад).



Двухфазный счетчик

Двухфазный счетчик выполняет счет:

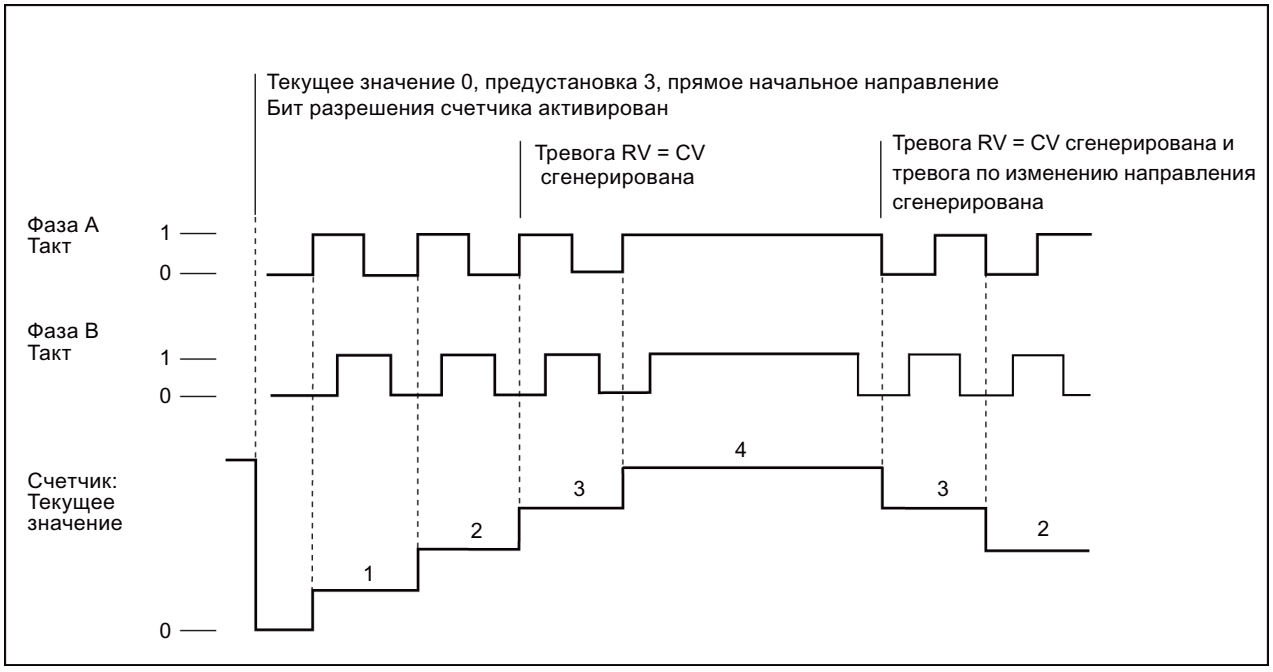
- Вперед для входа сигналов прямого счёта
- Назад для входа сигналов обратного счёта



A/B-счетчик

A/B-счетчик выполняет счет:

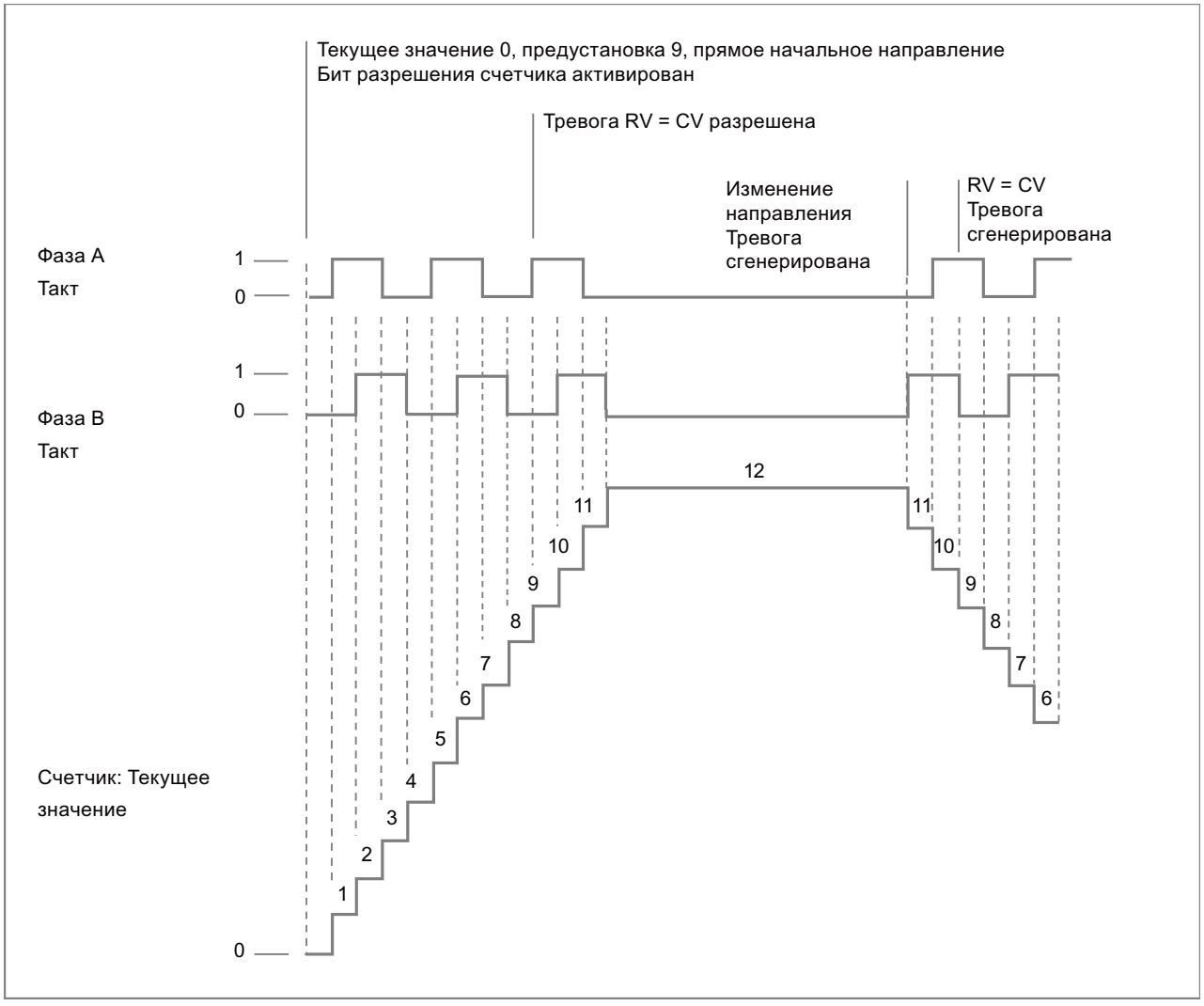
- Вперед при положительном фронте на счетном входе A, если счетный вход B имеет низкий уровень (Low)
- Назад при отрицательном фронте на счетном входе A, если счетный вход B имеет низкий уровень (Low)



Квадратурный A/B-счетчик

Квадратурный A/B-счетчик считает:

- Вперед при положительном фронте на счетном входе A, если счетный вход B имеет низкий уровень (Low)
- Вперед при отрицательном фронте на счетном входе A, если счетный вход B имеет высокий уровень (High)
- Вперед при положительном фронте на счетном входе B, если счетный вход A имеет высокий уровень (High)
- Вперед при отрицательном фронте на счетном входе B, если счетный вход A имеет низкий уровень (Low)
- Назад при положительном фронте на счетном входе B, если счетный вход A имеет низкий уровень (Low)
- Назад при отрицательном фронте на счетном входе B, если счетный вход A имеет высокий уровень (High)
- Назад при положительном фронте на счетном входе A, если счетный вход B имеет высокий уровень (High)
- Назад при отрицательном фронте на счетном входе A, если счетный вход B имеет низкий уровень (Low)



10.1.3.3 Начальные значения

При каждом переходе CPU в режим RUN, он загружает начальные значения. Начальные значения используются только в счетном режиме работы:

- Начальное значение счетчика: Программа устанавливает текущее значение счетчика на начальное значение счетчика, когда CPU переходит из режима STOP в режим RUN или когда программа запускает вход синхронизации.
- Начальное опорное значение: Когда текущее значение счетчика достигает опорного значения, и если соответствующие функции установлены, программа генерирует прерывание и / или выходной импульс.
- Начальное опорное значение 2: Если текущее значение счетчика достигает опорного значения 2, и если функция установлена, программа генерирует выходной импульс.

- Начальное верхнее предельное значение: максимальное значение счета. Значение по умолчанию - это максимально возможное значение +2 147 483 647 импульсов.
- Начальное нижнее предельное значение: минимальное значение счета. Значение по умолчанию - это минимальное возможное значение -2 147 483 648 импульсов.

Указанные выше значения и поведение счетчика при достижении предельного значения доступны только в режиме подсчета. Можно настроить значения и поведение с помощью инструкции CTRL_HSC_EXT, используя HSC_Count SDT.

10.1.3.4 Функции входов

Входы синхросигнала и направления определяют события и направление подсчета на основе рабочей фазы. В режиме счета можно использовать только входы синхронизации, захвата и ограничителя счета и можно отдельно включать и настраивать входы для различных типов запускающих событий.

Вход синхронизации

Вход синхронизации устанавливает текущее значение счета на стартовое значение (начальное значение счетчика). Вход синхронизации обычно используется для сброса счетчика на 0. Синхронизация может быть запущена при нахождении входного соединителя в одном из следующих состояний:

- High
- Low
- Переход из Low на High
- Переход из High на Low
- Переход из High на Low или из Low на High

Вход захвата

Вход захвата устанавливает снятое значение счета на значение счета, сохраненное в момент запуска входа захвата. Захват может быть запущен при нахождении входного соединителя в одном из следующих состояний:

- Переход из Low на High
- Переход из High на Low
- Переход из High на Low или из Low на High

Вход ограничителя счета

Вход ограничителя счета останавливает счет HSC. Ограничитель счета может быть запущен при нахождении входного соединителя в одном из следующих состояний:

- High
- Low

10.1.3.5 Функция выхода

Функция выхода сравнения является единственной функцией для HSC и доступна только в режиме счета.

Выход сравнения

Можно сконфигурировать выход сравнения для генерации одиночного импульса при наступлении одного из следующих событий:

- Счетчик соответствует опорному значению (прямое направление счета)
- Счетчик соответствует опорному значению (обратное направление счета)
- Счетчик соответствует опорному значению (прямое или обратное направление счета)
- Счетчик соответствует опорному значению 2 (прямое направление счета)
- Счетчик соответствует опорному значению 2 (обратное направление счета)
- Счетчик соответствует опорному значению 2 (прямое или обратное направление счета)
- Положительное переполнение
- Отрицательное переполнение

Можно настроить выходные импульсы с временем цикла в диапазоне от 1 до 500 мс; время цикла по умолчанию составляет 10 мс. Можно установить ширину импульса или продолжительность включения в диапазоне от 1 до 100%; длительность импульса по умолчанию составляет 50%.

Если в течение указанного времени цикла происходит несколько событий выхода сравнения, выходные импульсы этих событий теряются из-за того, что текущий импульс еще не завершил свой цикл. После окончания импульса (по истечении заданного времени цикла) генератор импульсов готов генерировать новый импульс.

10.1.3.6 Прерывающие события

В области конфигурации события можно выбрать OB аппаратного прерывания из выпадающего меню (или создать новый OB) и назначить его событию HSC. Приоритет прерывания находится в диапазоне от 2 до 26, где 2 - самый низкий приоритет, а 26 - самый высокий приоритет. В зависимости от конфигурации HSC, доступны следующие события:

- Событие "Значение счетчика соответствует опорному значению": Событие "значение счетчика соответствует опорному значению" возникает тогда, когда значение счетчика HSC достигает опорного значения. Опорное значение может быть установлено при конфигурировании в разделе исходного опорного значения или путем обновления значения "NewReference1" с использованием инструкции CTRL_HSC_EXT. Обратиться к разделу "Рабочая фаза (Страница 576)" для получения дополнительной информации.
- Событие синхронизации: Синхронизация выполняется всегда после активации и запуска входа синхронизации.
- Событие изменения направления: Событие изменения направления наступает при изменении направления счета. Обратиться к разделу "Рабочая фаза (Страница 576)" для получения дополнительной информации.

10.1.3.7 Разводка соединителей аппаратного входа

Для каждого включаемого входа HSC следует выбрать нужную точку входа, расположенную либо на CPU, либо на дополнительной сигнальное плате (коммуникационные и сигнальные модули не поддерживают входы HSC). При выборе входа, STEP 7 отображает максимальное значение частоты рядом с выбором. Может потребоваться настройка параметров цифрового входного фильтра, чтобы все частоты сигнала могли проходить через фильтр. Для настройки входных фильтров HSC см. "Настройка времени фильтра цифровых входов (Страница 169)".

Примечание

Входные каналы CPU и SB (от версии прошивки V4) имеют конфигурируемое время фильтра для входа

В старых версиях прошивки использовались фиксированные входные каналы HSC и время фильтра, которые не могли быть изменены.

От версии V4 можно устанавливаться входные каналы и время фильтра. Установка по умолчанию для фильтра для входа составляет 6,4 мс, что ограничивает макс. частоту счета до 78 Гц. Можно изменить установки фильтра, чтобы, в зависимости от конфигурации системы, считать более низкие и высокие частоты.

⚠ ПРЕДУПРЕЖДЕНИЕ

Риски, связанные с изменениями времени фильтра для каналов цифрового ввода

Если время фильтра для канала цифрового ввода изменяется по сравнению с предыдущей настройкой, то для нового входного значения может потребоваться удержание уровня "0" в течение до 20,0 мс, чтобы фильтр мог полноценно реагировать на новые входы. В течение этого времени может случиться, что короткие импульсы с уровнем "0" продолжительностью менее 20,0 мс не смогут быть обнаружены или подсчитаны.

Изменение времени фильтра может вызвать непредсказуемое поведение оборудования или процесса и, как следствие, привести к тяжким или опасным для жизни телесным повреждениям и/или материальному ущербу.

Чтобы гарантировать, что новое время фильтра сразу вступит в силу, необходимо выключить и снова включить CPU.

Использовать следующую таблицу и проследить, чтобы подключенные входные каналы CPU и SB поддерживали максимальные частоты импульсов в технологических сигналах:

Таблица 10-9 Вход CPU: Максимальная частота

CPU	Входной канал CPU	Рабочая фаза: Однофазный счетчик или двухфазный счетчик	Рабочая фаза: A/B-счетчик или квадратурный A/B-счетчик
1211C	Ia.0 до Ia.5	100 кГц	80 кГц
1212C	Ia.0 до Ia.5	100 кГц	80 кГц
	Ia.6, Ia.7	30 кГц	20 кГц
1214C и 1215C	Ia.0 до Ia.5	100 кГц	80 кГц
	Ia.6 до Ib.5	30 кГц	20 кГц
1217C	Ia.0 до Ia.5	100 кГц	80 кГц
	Ia.6 до Ib.1	30 кГц	20 кГц
	Ib.2 до Ib.5 (0,2+, 0,2- до 0,5+, 0,5-)	1 МГц	1 МГц

Таблица 10-10 Вход SB: Максимальная частота (опциональная плата)

Сигнальная плата (SB)	Входной канал SB	Рабочая фаза: Однофазный счетчик или двухфазный счетчик	Рабочая фаза: A/B-счетчик или квадратурный A/B-счетчик
SB 1221, 200 кГц	Ie.0 до Ie.3	200 кГц	160 кГц
SB 1223, 200 кГц	Ie.0, Ie.1	200 кГц	160 кГц
SB 1223	Ie.0, Ie.1	30 кГц	20 кГц

При назначении точки входа для функции HSC, можно назначить одну и ту же точку входа нескольким функциям HSC. Пример: Назначение I0.3 для входа синхронизации HSC1 и входа синхронизации HSC2 для синхронизации счета обоих HSC одновременно является допустимой конфигурацией. Однако она генерирует предупреждение при компиляции.

По возможности избегать назначения нескольких входных функций одного и того же HSC одной и той же точке входа. Пример: Назначение I0.3 для входа синхронизации и входа ограничителя счета HSC 1 для синхронизации счета и одновременно для отключения счета также является допустимой конфигурацией. Можно использовать такую конфигурацию, но она может привести к непредвиденным результатам.

⚠ ПРЕДУПРЕЖДЕНИЕ

Риски, связанные с назначением нескольких функций одному каналу цифрового ввода

Присвоение нескольких входных функций одного HSC общей точке входа может привести к непредсказуемым результатам. Когда запуск происходит на входе с несколькими назначенными функциями, порядок, в котором функции выполняются PLC, не может быть определен. Это известно как состояние гонки и часто является нежелательной ситуацией.

Такое состояние гонки может вызвать непредсказуемое поведение оборудования или процесса и, как следствие, привести к тяжким или опасным для жизни телесным повреждениям и/или материальному ущербу.

Чтобы избежать состояния гонки, не назначать более двух входных функций одного и того же HSC одному и тому же входному соединителю. Если HSC имеет две входные функции, назначенные одному и тому же соединителю, установить пусковые события таким образом, чтобы они никогда не возникали одновременно. Следует помните, что отрицательный фронт возникает в тот же момент, когда начинается низкий уровень, и что положительный фронт возникает в тот же момент, когда начинается высокий уровень.

Примечание

Цифровые входы и выходы, используемые высокоскоростными счетчиками (HSC), назначаются во время конфигурации устройства CPU. При назначении входов и выходов для устройств HSC, нельзя изменять значения этих входов и выходов с помощью функции принудительного присваивания значений в таблице текущего контроля. HSC полностью контролирует эти входы и выходы.

10.1.3.8 Разводка соединителей аппаратного выхода

Если активируется выход сравнения, то следует выбрать доступный выход. После конфигурирования выхода для использования HSC (или другими технологическими объектами, такими как генератор импульсов), этот выход принадлежит исключительно этому объекту. Никакой другой компонент не может использовать выход, и выход не может быть принудительно установлен на значение. Если один единственный выход конфигурируется для нескольких HSC или для использования в HSC и импульсном выходе, программа генерирует ошибку компиляции.

10.1.3.9 Адреса ввода HSC

Каждый HSC использует двойное слово в области I памяти, в которой хранится текущее значение счета. Если HSC конфигурируется для режима частоты, то частота сохраняется по этому адресу ввода. Доступный диапазон адресов ввода - от I0.0 до I1023.7 (максимальный начальный адрес - I1020.0). HSC не может использовать адрес ввода, который пересекается с адресом ввода, назначенным другому компоненту. Дополнительную информацию об образце процесса см. "Выполнение программы пользователя (Страница 83)".

В следующей таблице представлены стандартные адреса для каждого HSC:

Таблица 10-11 Стандартные адреса HSC

Высокоскоростной счётчик (HSC)	Тип данных актуального значения	Стандартный адрес актуального значения
HSC1	DInt	ID 1000
HSC2	DInt	ID 1004
HSC3	DInt	ID 1008
HSC4	DInt	ID 1012
HSC5	DInt	ID 1016
HSC6	DInt	ID 1020

10.1.3.10 Аппаратный идентификатор

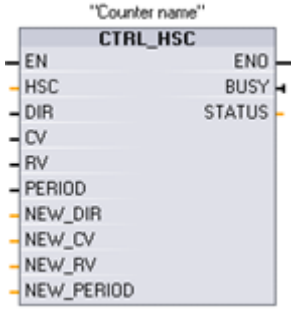
Каждый HSC имеет уникальный аппаратный идентификатор, который используется инструкциями HSC_CTRL и HSC_CTRL_EXT. Переменная (тег) PLC для аппаратного идентификатора оборудования находится в разделе "Системные константы". HSC с именем "HSC_1" имеет переменную (тег) «Local ~ HSC_1» и тип данных "Hw_Hsc". Эта переменная (тег) также отображается в раскрывающемся меню при выборе входа HSC для инструкций CTRL_HSC_EXT.

10.1.4 Старая инструкция CTRL_HSC (управление высокоскоростным счетчиком)

10.1.4.1 Обзор инструкции

Таблица 10-12 Инструкция CTRL_HSC (для общих заданий счета)

LAD/FBD	SCL	Описание

	`"CTRL_HSC_1_DB" (`
	`hsc:=W#16#0,`
	`dir:=False,`
	`cv:=False,`
	`rv:=False,`
	`period:=False,`
	`new_dir:=0,`
	`new_cv:=L#0,`
	`new_rv:=L#0,`
	`new_period:=0,`
	`busy=>_bool_out_,`
	`status=>_word_out_);`
		Для сохранения данных счетчика, каждая инструкция CTRL_HSC (управление высокоскоростным счетчиком) использует сохраненную в DB структуру. При добавлении инструкции CTRL_HSC в редакторе, необходимо назначить блок данных.

[1] При вставке инструкции STEP 7 открывает диалоговое окно "Опции вызова" для создания соответствующего DB.

[2] В примере SCL "CTRL_HSC_1_DB" - это имя DB экземпляра.

Таблица 10-13 Типы данных для параметров

Параметр	Объявление	Тип данных	Описание
HSC	IN	HW_HSC	Идентификатор HSC
DIR[1, 2]	IN	Bool	1 = Запрос нового направления
CV[1]	IN	Bool	1 = Запрос на установку нового значения счета
RV[1]	IN	Bool	1 = Запрос на установку нового опорного значения
PERIOD[1]	IN	Bool	1 = Запрос на установку нового интервала времени (только для измерения частоты)
NEW_DIR	IN	Int	Новое направление: 1= вперед, -1= назад
NEW_CV	IN	DInt	Новое значение счета
NEW_RV	IN	DInt	Новое опорное значение
NEW_PERIOD	IN	Int	Новое значение для интервала времени указывается в миллисекундах (только для измерения частоты). Допускаются только значения 10, 100 или 1000 мс: 1000 = 1 секунда 100 = 0,1 секунды 10 = 0,01 секунды

Параметр	Объявление	Тип данных	Описание
BUSY[3]	OUT	Bool	Функция занята
STATUS	OUT	Word	Условие выполнения

[1] Если обновление значения для параметра не запрашивается, то соответствующие входные значения игнорируются.

[2] Параметр DIR действителен только в том случае, если для сконфигурированного направления счета задано программное управление (внутреннее управление направлением). Пользователь определяет, как этот параметр будет использоваться в конфигурации устройства HSC.

[3] Для HSC в CPU или на SB параметр BUSY всегда значение 0.

Для конфигурирования параметров для каждого HSC используется конфигурация устройства CPU: счетная/частотная функция, опции сброса, конфигурация прерывающего события, аппаратные входы/выходы и адрес счетного значения.

Некоторые из параметров для HSC могут изменяться из программы пользователя для установки режима программного управления для процесса счета:

- Установка направления счета на значение NEW_DIR
- Установка текущего значения счета на новое значение NEW_CV
- Установка опорного значения на новое значение NEW_RV
- Установка значения интервала времени (только для измерения частоты) на новое значение NEW_PERIOD

Если следующие значения логического флага установлены на 1 при выполнении инструкции CTRL_HSC, то соответствующее значение NEW_xxx загружается в счетчик. Несколько запросов (более одного флага установлено одновременно) обрабатываются за одно выполнение инструкции CTRL_HSC.

- DIR = 1 - это запрос на загрузку значения NEW_DIR, 0 = изменения отсутствуют
- CV = 1 - это запрос на загрузку значения NEW_CV, 0 = изменения отсутствуют
- RV = 1 - это запрос на загрузку значения NEW_RV, 0 = изменения отсутствуют
- PERIOD = 1 - это запрос на загрузку значения NEW_PERIOD, 0 = изменения отсутствуют

При ошибке ENO устанавливается на 0 и выход STATUS возвращает код условия:

Таблица 10-14 Execution condition codes

STATUS (W#16#)	Описание
0	Ошибки отсутствуют
80A1	Идентификатор HSC не адресует HSC
80B1	Недействительное значение в NEW_DIR
80B2	Недействительное значение в NEW_CV
80B3	Недействительное значение в NEW_RV
80B4	Недействительное значение в NEW_PERIOD

STATUS (W#16#)	Описание
80C0	Многократное обращение к высокоскоростному счетчику Эта ошибка может возникнуть, если для типа счета (Страница 575) установлен "Период" или "Управление перемещением". Эти типы счета не работают для инструкции CTRL_HSC и поддерживаются только инструкцией CTRL_HSC_EXT.
80D0	Высокоскоростной счётчик (HSC) не активирован в конфигурации оборудования CPU

10.1.4.2 Использование CTRL_HSC

Инструкция CTRL_HSC обычно помещается в OB аппаратных прерываний, который выполняется, когда запускается событие аппаратного прерывания для счетчика. Например, если событие CV = RV запускает прерывание счетчика, то OB аппаратных прерываний выполняет инструкцию CTRL_HSC и может изменить опорное значение, загрузив значение NEW_RV.

Текущее значение счетчика недоступно в параметрах для CTRL_HSC. Адрес образа процесса, в котором хранится текущее значение счетчика, назначается во время конфигурации оборудования высокоскоростному счетчику. Можно использовать программную логику для прямого считывания значения счетчика. Значение, возвращаемое программе, будет правильным счетом на момент считывания счетчика. Счетчик продолжит считать высокоскоростные события. Поэтому фактическое значение счетчика может измениться до того, как программа завершит процесс, используя старое значение счетчика.

10.1.4.3 Текущее состояние счетчика HSC

CPU хранит текущее значение каждого HSC по адресу входа (I). В следующей таблице представлены стандартные адреса для текущего значения каждого HSC. Можно изменить адрес I для текущего значения, изменив свойства CPU в конфигурации устройства.

Высокоскоростные счетчики используют значение DInt для хранения текущего значения счетчика. Значение счетчика DInt имеет диапазон от -2147483648 до +2147483647. Начиная с прошивки CPU V4.2, можно выбирать пределы диапазона. Дополнительную информацию см. в: "Начальные значения (Страница 580)".

Счетчик переходит от максимального положительного значения к максимальному отрицательному значению при прямом счете и от максимального отрицательного значения к максимальному положительному значению при обратном счете. Частота возвращается в Герцах (пример: 123,4 Гц возвращается как 123).

Таблица 10-15 Стандартные адреса HSC

HSC	Тип данных актуального значения	Стандартный адрес актуального значения
HSC1	DInt	ED1000
HSC2	DInt	ED1004
HSC3	DInt	ED1008

HSC	Тип данных актуального значения	Стандартный адрес актуального значения
HSC4	DInt	ED1012
HSC5	DInt	ED1016
HSC6	DInt	ED1020

10.2 ПИД-регулирование

STEP 7 предлагает следующие ПИД инструкции для S7-1200 CPU:

- Инструкция PID_Compact служит для регулирования технических процессов с непрерывными входными и выходными переменными.
- Инструкция PID_3Step используется для управления устройствами с электроприводом, такими как клапаны, которые требуют цифровых сигналов для открытия и закрытия.
- Инструкция PID_Temp предоставляет универсальный ПИД-регулятор для специальных требований при регулировании температуры.

Примечание

Изменения, вносимые в конфигурацию ПИД и загружаемые в рабочем состоянии RUN в целевую систему, не вступают в силу, пока CPU не перейдет из режима STOP в режим RUN. Изменения, вносимые в диалоговом окне "Параметры ПИД" с помощью "Управление стартовым значением", вступают в силу немедленно.

Все три ПИД инструкции (PID_Compact, PID_3Step иPID_Temp) могут рассчитать П-, И- и Д-составляющую при запуске (если сконфигурирована "предварительная настройка"). Также инструкция может быть сконфигурирована для "точной регулировки" с оптимизацией параметров. Не требуется определять параметры вручную.

Примечание

Выполнение ПИД инструкции с постоянными повторяющимися интервалами времени выборки (по возможности в ОВ программного цикла).

Поскольку ПИД-регулятору требуется определенное время для реагирования на изменения значения регулирующего воздействия, не следует рассчитывать выходное значение в каждом цикле. Не выполнять ПИД инструкцию в ОВ программного цикла главной программы (например, ОВ 1).

Время выборки ПИД алгоритма представляет собой время между двумя расчетами выходного значения (значения регулирующего воздействия). Выходное значение рассчитывается во время автонастройки и округляется до кратного времени цикла. Все остальные функции ПИД инструкции выполняются при каждом вызове.

ПИД алгоритм

ПИД (Пропорциональный/Интегральный/Дифференциальный) регулятор измеряет временной интервал между двумя вызовами и затем анализирует результаты для контроля времени выборки. Среднее значение времени выборки генерируется при каждом переключении режима и во время начального запуска. Это значение используется в качестве опорного значения для контролирующей функции и для вычисления. Контроль включает в себя текущее время измерения между двумя вызовами и среднее значение определенного времени выборки контроллера.

Выходное значение ПИД-регулятора включает в себя три составляющие:

- P (пропорциональная): При расчете с П-составляющей, выходное значение пропорционально разнице между заданным и технологическим значением (входным значением).
- И (интегральная): При расчете с И-составляющей, выходное значение увеличивается пропорционально длительности разницы между заданным и технологическим значением (входным значением) для исправления расхождения.
- Д (дифференциальная): При расчете с Д-составляющей, выходное значение увеличивается как функция растущей скорости изменения разницы между заданным и технологическим значением (входным значением). Выходное значение уравнивается к заданному значению с максимально возможной скоростью.

ПИД-регулятор рассчитывает выходное значение для инструкции PID_Compact по следующей формуле.

$$y = K_p \left[(b \cdot w - x) + \frac{1}{T_I \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

y	Выходное значение	x	Технологическое значение
w	Заданное значение	s	Оператор Лапласа
K_p	Пропорциональное усиление (П-составляющая)	a	Коэффициент для задержки дифференцирования (Д-составляющая)
T_1	Время интегрирования (И-составляющая)	b	Коэффициент эквивалентности пропорциональной составляющей (П-составляющей)
T_D	Время дифференцирования (Д-составляющая)	c	Коэффициент эквивалентности дифференциальной составляющей (Д-составляющей)

ПИД-регулятор рассчитывает выходное значение для инструкции PID_3Step по следующей формуле.

$$\Delta y = K_p \cdot s \cdot \left[(b \cdot w - x) + \frac{1}{T_I \cdot s} (w - x) + \frac{T_D \cdot s}{a \cdot T_D \cdot s + 1} (c \cdot w - x) \right]$$

y	Выходное значение	x	Технологическое значение
w	Заданное значение	s	Оператор Лапласа
K_p	Пропорциональное усиление (П-составляющая)	a	Коэффициент для задержки дифференцирования (Д-составляющая)
T_1	Время интегрирования (И-составляющая)	b	Коэффициент эквивалентности пропорциональной составляющей (П-составляющей)
T_D	Время дифференцирования (Д-составляющая)	c	Коэффициент эквивалентности дифференциальной составляющей (Д-составляющей)

10.2.1 Вставка ПИД инструкции и технологического объекта

STEP 7 предлагает две инструкции для ПИД-регулятора:

- Инструкция PID_Compact с соответствующим ТО предлагает универсальный ПИД-регулятор с настройкой. Технологический объект содержит все настройки контура регулирования.
- Инструкция PID_3Step с соответствующим ТО предлагает ПИД-регулятор с определенными настройками для моторизованных вентилей. Технологический объект содержит все настройки контура регулирования. Регулятор PID_3Step предлагает два дополнительных булевых выхода.

После создания ТО необходимо сконфигурировать параметры (Страница 626). Также необходимо согласовать параметры для автоматической настройки ("Предварительная настройка" во время запуска или ручная "Точная регулировка") для ввода ПИД-регулятора в эксплуатацию (Страница 644).

Таблица 10-16 Вставка ПИД инструкции и технологического объекта

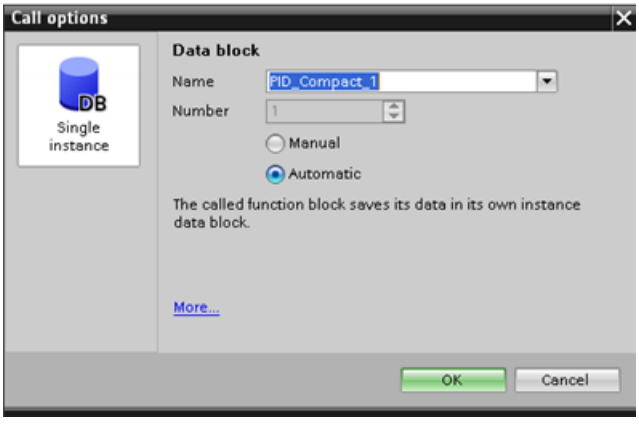
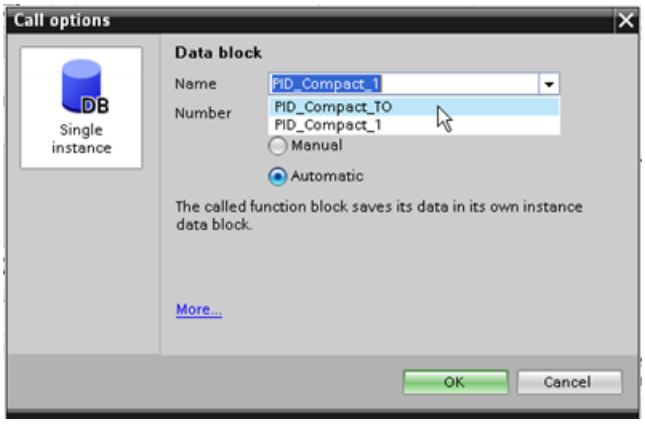
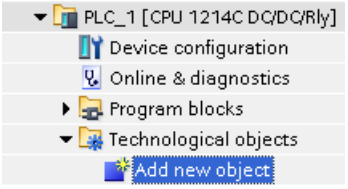
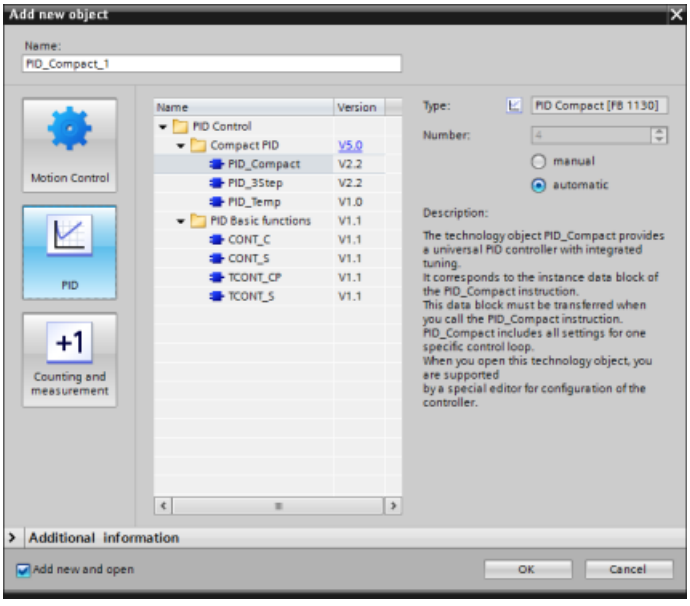
При вставке ПИД инструкции в программу пользователя, STEP 7 автоматически создает технологический объект и DB экземпляра для инструкции. DB экземпляра содержит все параметры для ПИД инструкции. Для правильной работы каждой ПИД инструкции требуется собственный, однозначный DB экземпляра. После вставки ПИД инструкции и создания ТО, а также DB экземпляра, следует сконфигурировать параметры для технологического объекта (Страница 626).	

Таблица 10-17 (Опция) Создание ТО в дереве проекта

Технологические объекты для проекта могут создаваться и **до** вставки ПИД инструкции. Если технологический объект создается до вставки ПИД инструкции в программе пользователя, технологический объект может быть выбран позднее на этапе вставки ПИД инструкции.	
Для выбора ТО выполнить двойной клик по символу "Добавить новый объект" в дереве проекта.	
Кликнуть по символу "Регулирование" и выбрать ТО для типа ПИД-регулятора (PID_Compact или PID_3Step). Также можно ввести имя для технологического объекта. Нажать "OK" для создания ТО.	

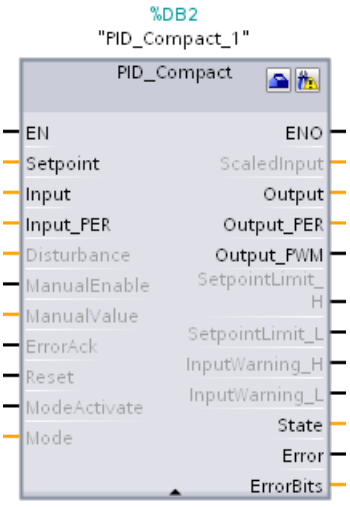
10.2.2 PID_Compact

10.2.2.1 Инструкция PID_Compact

Инструкция PID_Compact предлагает универсальный ПИД-регулятор со встроенной автоматической настройкой для автоматического и ручного режима.

Таблица 10-18 Инструкция PID_Compact

LAD/FBD	SCL	Описание

	`"PID_Compact_1"(`
	`  Setpoint:=_real_in_,`
	`  Input:=_real_in_,`
	`  Input_PER:=_word_in_,`
	`  Disturbance:=_real_in_,`
	`  ManualEnable:=_bool_in_,`
	`  ManualValue:=_real_in_,`
	`  ErrorAck:=_bool_in_,`
	`  Reset:=_bool_in_,`
	`  ModeActivate:=_bool_in_,`
	`  Mode:=_int_in_,`
	`  ScaledInput=>_real_out_,`
	`  Output=>_real_out_,`
	`  Output_PER=>_word_out_,`
	`  Output_PWM=>_bool_out_,`
	
	`SetpointLimit_H=>_bool_out_,`
	
	`SetpointLimit_L=>_bool_out_,`
	`  InputWarning_H=>_bool_out_,`
	`  InputWarning_L=>_bool_out_,`
	`  State=>_int_out_,`
	`  Error=>_bool_out_,`
	`  ErrorBits=>_dword_out_);`
		PID_Compact предлагает ПИД-регулятор с автоматической настройкой для автоматического и ручного режима PID_Compact - это PIDT1 регулятор с анти-насыщением и оценкой П- и Д-составляющей.

[1] При добавлении инструкции, STEP 7 автоматически создает TO и DB экземпляра. DB экземпляра содержит параметры регулятора TO.

[2] В примере SCL "PID_Compact_1" - это имя DB экземпляра.

Таблица 10-19 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
Setpoint	IN	Real	Заданное значение ПИД-регулятора в автоматическом режиме (значение по умолчанию: 0,0)
Input	IN	Real	Переменная (тег) программы пользователя используется в качестве источника технологического значения. (значение по умолчанию: 0,0) При использовании параметра Input, требуется установка Config.InputPerOn = FALSE.

Параметр и тип данных		Тип данных	Описание
Input_PER	IN	Word	Аналоговый вход используется в качестве источника технологического значения. (значение по умолчанию: W#16#0) При использовании параметра Input_PER, требуется установка Config.InputPerOn = TRUE.
Disturbance	IN	Real	Возмущающее воздействие или значение предварительного регулирования
ManualEnable	IN	Bool	Включает или выключает ручной режим. (значение по умолчанию: FALSE): • Изменение фронта с FALSE на TRUE активирует "Ручной режим работы", в то время как в состоянии = 4 режим работы не изменяется. Пока ManualEnable = TRUE, режим работы не может переключаться по положительному фронту на ModeActivate, также нельзя использовать диалоговое окно для ввода в эксплуатацию. • Изменение фронта с TRUE на FALSE активирует назначенный Mode режим работы. Примечание: Изменять режим работы рекомендуется через ModeActivate.
ManualValue	IN	Real	Выходное значение для ручного режима. (значение по умолчанию: 0,0) Допустимые значения: от Config.OutputLowerLimit до Config.OutputUpperLimit.
ErrorAck	IN	Bool	Сбрасывает ErrorBits и выход предупреждения. Изменение фронта с FALSE на TRUE
Reset	IN	Bool	Перезапускает регулятор. (значение по умолчанию: FALSE): • Изменение фронта с FALSE на TRUE: – Выполняет переключение в "неактивный" режим – Сбрасывает ErrorBits и выходы предупреждения. – Удаляет интегральную составляющую – Сохраняет параметры ПИД • Пока Reset = TRUE, PID_Compact остается в "неактивном" режиме (состояние = 0). • Изменение фронта с TRUE на FALSE: – PID_Compact переходит в режим работы, сохраненный в параметре Mode.
ModeActivate	IN	Bool	PID_Compact переключается в рабочее состояние, сохраненное в параметре Mode. Изменение фронта с FALSE на TRUE:
Mode	IN	Int	Требуемый режим работы ПИД; будет активироваться положительным фронтом на входе Mode Activate .
ScaledInput	OUT	Real	Масштабированное технологическое значение. (значение по умолчанию: 0,0)
Output[1]	OUT	Real	Выходное значение в формате REAL. (значение по умолчанию: 0,0)
Output_PER[1]	OUT	Word	Аналоговое выходное значение. (значение по умолчанию: W#16#0)
Output_PWM[1]	OUT	Bool	Выходное значение для широтно-импульсной модуляции. (значение по умолчанию: FALSE) Выходное значение формируется из времени включения и выключения.

Параметр и тип данных		Тип данных	Описание
SetpointLimit_H	OUT	Bool	Верхнее предельное заданное значение. (значение по умолчанию: FALSE) Если SetpointLimit_H = TRUE, то абсолютное верхнее предельное значение заданного значения достигнуто (Setpoint ≥ Config.SetpointUpperLimit). Заданное значение ограничено до Config.SetpointUpperLimit.
SetpointLimit_L	OUT	Bool	Нижнее предельное заданное значение. (значение по умолчанию: FALSE) Если SetpointLimit_L = TRUE, то абсолютное нижнее предельное значение заданного значения достигнуто (Setpoint ≥ Config.SetpointLowerLimit). Заданное значение ограничено до Config.SetpointLowerLimit.
InputWarning_H	OUT	Bool	Если InputWarning_H = TRUE, то технологическое значение достигло или превысило верхнюю границу предупреждения. (значение по умолчанию: FALSE)
InputWarning_L	OUT	Bool	Если InputWarning_L = TRUE, то технологическое значение достигло или перешло нижнюю границу предупреждения. (значение по умолчанию: FALSE)
State	OUT	Int	Текущий режим работы ПИД-регулятора. (значение по умолчанию: 0) Изменить режим работы можно при помощи входного параметра Mode и положительного фронта на ModeActivate: • State = 0: Неактивно • State = 1: Предварительная настройка • State = 2: Ручная точная регулировка • State = 3: Автоматический режим • State = 4: Ручной режим • State = 5: Замещающее выходное значение с контролем ошибок
Error	OUT	Bool	Если Error = TRUE, значит в этом цикле активно как минимум одно сообщение об ошибке. (значение по умолчанию: FALSE) Примечание: Параметр Error в V1.x ПИД было полем ErrorBits с кодами ошибок. Теперь булевский флаг является указанием на то, что возникла ошибка.
ErrorBits	OUT	DWord	В таблице параметров (Страница 599) ErrorBits для инструкции PID_Compact описываются активные сообщения об ошибках. (значение по умолчанию: DW#16#0000 (ошибки отсутствуют)). Параметр ErrorBits является сохраняющимся или сбрасывается по положительному фронту на Reset или ErrorAck. Примечание: В V1.x параметр ErrorBits был определен как параметр Error и отсутствовал.

[1] Выходы параметров Output, Output_PER и Output_PWM могут использоваться параллельно.

Принцип работы регулятора PID_Compact

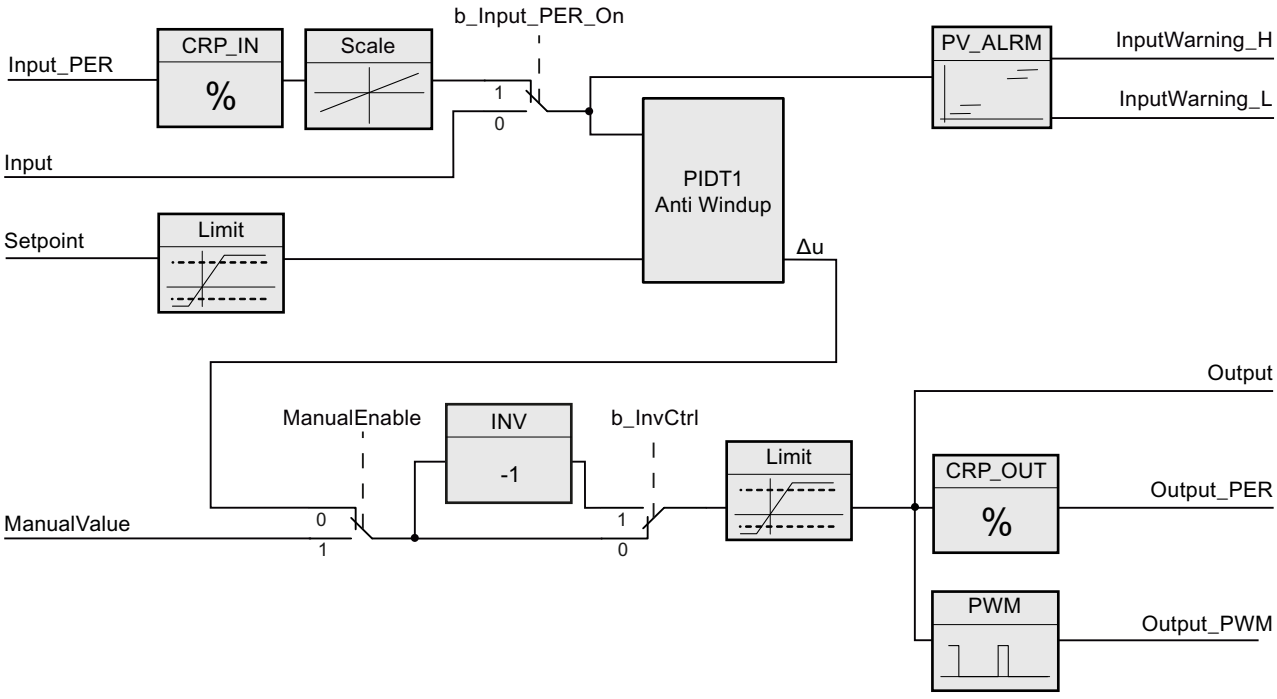


Рисунок 10-1 Принцип работы регулятора PID_Compact

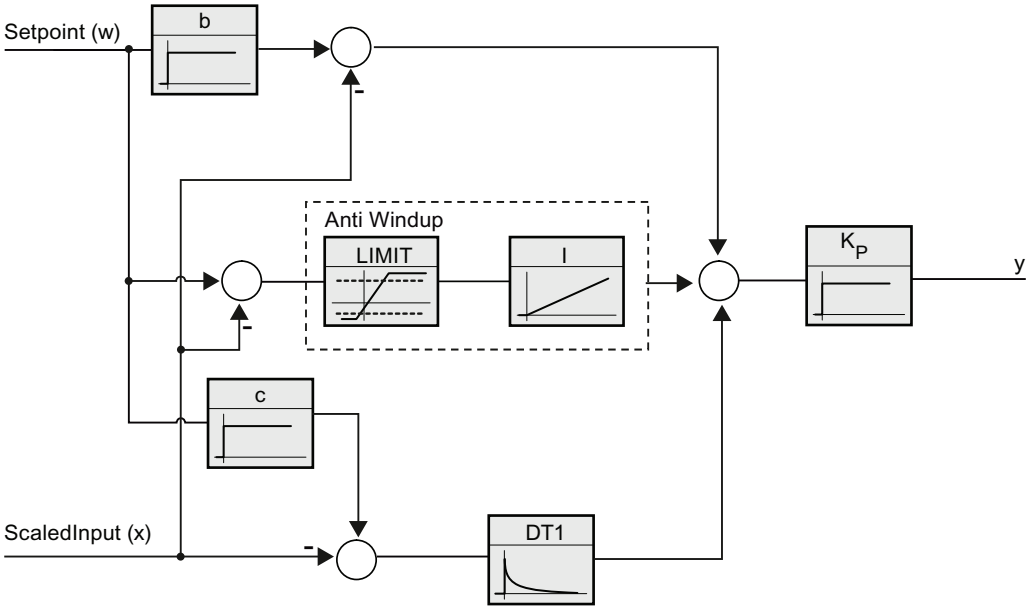


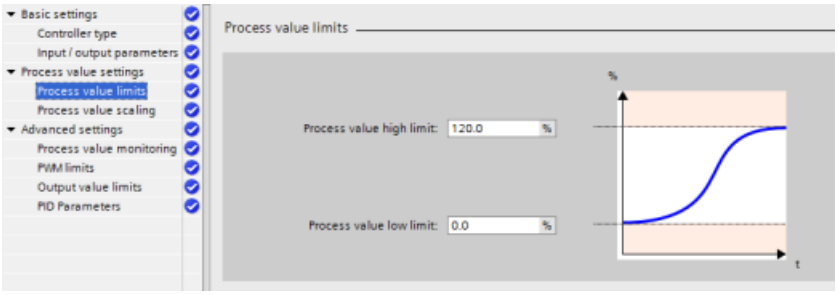
Рисунок 10-2 Принцип работы регулятора PID_Compact как PIDT1 регулятора с анти-насыщением

10.2.2.2 Предельные значения для технологического значения инструкции PID_Compact

"Предельные значения для технологического значения" обычно применяются в комбинации с аналоговым входом, но могут использоваться и для других целей.

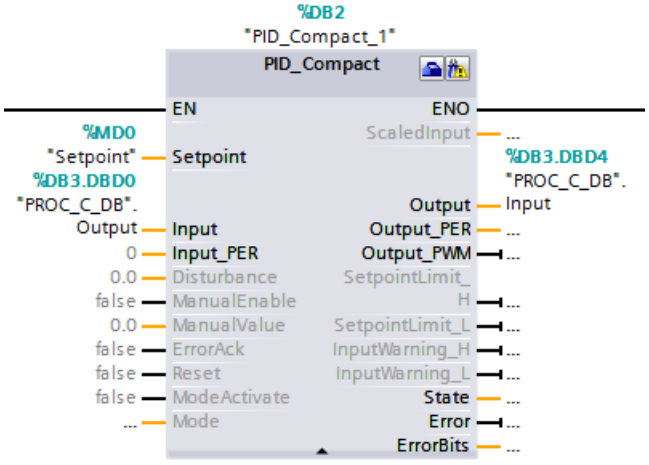
Конфигурация предельных значений для технологического значения служит двум целям:

- Установка выходов аварийных сигналов для верхнего и нижнего предельного значения на ПИД блоке.
- Установка предельного значения, чтобы переменная процесса не могла нарушить этой предельное значение, независимо от заданного значения. Эта конфигурация определяет фиксированные предельные значения для технологического процесса.



На ПИД блоке можно использовать четыре выхода аварийных сигналов:

- SetpointLimit_H: Заданное значение превышает верхнее предельное значение.
- SetpointLimit_L: Заданное значение нарушает нижнее предельное значение.
- InputWarning_H: Переменная процесса превышает верхнее предельное значение входа.
- InputWarning_L: Переменная процесса нарушает нижнее предельное значение входа.



При настройке пределов для технологических значений, входы/выходы, вызывающие состояние true для выходов аварийных сигналов по верхней и нижней границе заданных значений, должно иметь одинаковую установку. Например, с настройками, показанными выше, если ввести заданное значение, превышающее 120%, то выход "SetpointLimit_H" станет true. Это действие необходимо повторить и для нижнего предела. Если ввести заданное значение ниже 0%, то выход "SetpointLimit_L" станет true. Для программы это является указанием на то, что введенное заданное значение находится вне допустимого диапазона. Это аварийное сообщение может служить сигналом для повторного ввода заданного значения.

Если вводится заданное значение вне допустимого диапазона, Compact_PID автоматически ограничивает переменную процесса до настроенного диапазона. Например, если верхний предел технологического значения установлен на 120% (как показано на рисунке выше), все равно можно ввести заданное значение выше 120%. Когда переменная процесса приближается к 120%, ПИД понижает выход и регулирует процесс по верхнему пределу в 120%. И наоборот, то же действие происходит, если заданное значение ниже нижнего предела технологического значения. ПИД не позволяет переменной процесса опуститься ниже нижнего предела. Эта функция позволяет определить приемлемый технологический режим при штатном автоматическом ПИД регулировании. Однако эта функция доступна для запуска и выключения, только если ПИД находится в автоматическом режиме. Если ПИД находится в автоматическом режиме, а заданное значение и переменная процесса меньше нижнего предела, то ПИД пытается регулировать процесс по установленному нижнему пределу.

10.2.2.3 Параметр ErrorBit инструкции PID_Compact

Если активно несколько ошибок, то значения кодов ошибок отображаются с помощью двоичного сложения. Например, отображение кода ошибки 0003 указывает на то, что также активны ошибки 0001 и 0002.

Таблица 10-20 Параметр ErrorBit инструкции PID_Compact

ErrorBit (DW#16#...)	Описание
0000	Ошибки отсутствуют
0001 [1, 2]	Параметр Input находится вне допустимого диапазона технологических значений. Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 [2, 3]	Недействительное значение в параметре Input_PER. Проверить наличие ошибок на аналоговом входе.
0004 [4]	Ошибка при точной регулировке. Колебания технологического значения нельзя скомпенсировать.
0008 [4]	Ошибка при запуске предварительной настройки. Технологическое значение слишком приближено к заданному значению. Запустить точную регулировку.
0010 [4]	Заданное значение было изменено во время выполнения настройки. Примечание: Допустимые колебания на заданном значении могут быть установлены в переменной CancelTuningLevel.
0020	Предварительная настройка при точной регулировке не разрешена. Примечание: Если до возникновений ошибки имело место ActivateRecoverMode = TRUE, то PID_Compact остается в режиме точной регулировки.
0080 [4]	Ошибка при предварительной настройке. Неправильная конфигурация границ выходного значения. Проверить, правильно ли сконфигурированы границы выходного значения и соответствуют ли они логике регулирования.
0100 [4]	Ошибка при точной регулировке привела к недопустимым параметрам.

ErrorBit (DW#16#...)	Описание
0200 [2, 3]	Недействительное значение в параметре Input: Числовой формат значения недействителен.
0400 [2, 3]	Расчет выходного значения не удался. Проверить параметры ПИД.
0800 [1, 2]	Ошибка времени выборки: PID_Compact не вызывается при активном времени выборки OB циклического прерывания.
1000 [2, 3]	Недействительное значение в параметре Setpoint: Числовой формат значения недействителен.
10000	Недействительное значение в параметре ManualValue: Числовой формат значения недействителен. Примечание: Если до возникновений ошибки имело место ActivateRecoverMode = TRUE, то PID_Compact использует SubstituteOutput как выходное значение. Как только в ManualValue будет задано действительное значение, PID_Compact использует его в качестве выходного значения.
20000	Недопустимое значение переменной SubstituteValue: Числовой формат значения недействителен. PID_Compact использует нижнюю границу выходного значения в качестве выходного значения. Примечание: Если до ошибки был активен автоматический режим,ActivateRecoverMode = TRUE и ошибка больше неактивна, PID_Compact снова перейдет в автоматический режим.
40000	Недействительное значение в параметре Disturbance: Числовой формат значения недействителен. Примечание: Если до ошибки был активен автоматический режим и ActivateRecoverMode = FALSE, то Disturbance устанавливается на ноль. PID_Compact остается в автоматическом режиме. Примечание: Если до возникновения ошибки была активна предварительная настройка или точная регулировка и имело место ActivateRecoverMode = TRUE, то PID_Compact переходит в сохраненный в параметре Mode режим работы. Если ошибки на текущем этапе не влияют на выходное значение, то настройка не отменяется.

[1] Примечание: Если до ошибки был активен автоматический режим и ActivateRecoverMode = TRUE, то PID_Compact остается в автоматическом режиме.

[2] Примечание: Если до возникновения ошибки была активна предварительная настройка или точная регулировка и имело место ActivateRecoverMode = TRUE, то PID_Compact переходит в сохраненный в параметре Mode режим работы.

[3] Примечание: Если до ошибки был активен автоматический режим и ActivateRecoverMode = TRUE, PID_Compact возвращает сконфигурированное замещающее выходное значение. Когда ошибка будет устранена, PID_Compact снова переходит в автоматический режим.

[4] Примечание: Если до возникновения ошибки имело место ActivateRecoverMode = TRUE, то PID_Compact отменяет настройку и переходит в сохраненный в параметре Mode режим работы.

10.2.2.4 Параметры предупреждения инструкции PID_Compact

Если на ПИД-регуляторе активно несколько предупреждений, то значения кодов ошибок отображаются с помощью двоичного сложения. Например, отображение кода ошибки 0003 указывает на то, что также активны ошибки 0001 и 0002.

Таблица 10-21 Параметр Warning инструкции PID_Compact

Warning (DW#16#...)	Описание
0000	Предупреждения отсутствуют.
0001 [1]	Точка перегиба не была найдена во время предварительной настройки.
0002	При настройке в рабочем режиме были вызваны колебания. (Параметр "Warning" блокирует это предупреждение, оно доступно только в параметре “WarningInternal” для целей диагностики.)
0004 [1]	Заданное значение было ограничено до сконфигурированных пределов.
0008 [1]	Для выбранного методы вычисления не были определены все необходимые характеристики регулятора. В качестве замены параметры ПИД были рассчитаны при помощи метода TIR.TuneRuleHeat / TIR.TuneRuleCool = 3.
0010	Не удалось изменить рабочее состояние из-за Reset = TRUE или ManualEnable = TRUE.
0020	Время цикла вызывающего OB ограничивает время выборки ПИД алгоритма. Сокращение времени цикла OB может улучшить результаты.
0040 [1]	Технологическое значение превысило одну из границ предупреждения.
0080	Недопустимое значение в "Mode". Смена рабочего состояния выполнена не будет.
0100 [1]	Ручное значение было ограничено до предельных значений выхода регулятора.
0200	Указанное правило для настройки не поддерживается. Расчет параметров ПИД не выполняется.
1000	Замещающее выходное значение не может быть достигнуто, поскольку оно не соответствуют установленным для выходного значения пределам.

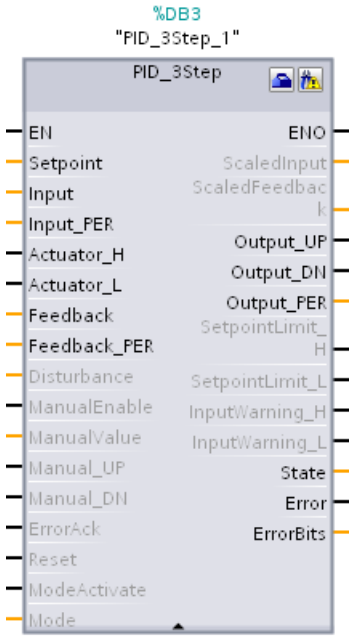
[1] Примечание: ПИД-регулятор, как только причина была устранена или реакция пользователя повторилась с действительными параметрами, автоматически удаляет следующие предупреждения: 0001, 0004, 0008, 0040 и 0100.

10.2.3 PID_3Step

10.2.3.1 Инструкция PID_3Step

Инструкция PID_3Step конфигурирует ПИД-регулятор с функциями автонастройки, оптимизированный для моторизованных вентилей и исполнительных устройств.

Таблица 10-22 Инструкция PID_3Step

LAD/FBD	SCL	Описание
	**"PID_3Step_1"(** **SetpoInt:=_real_in_,** **Input:=_real_in_,** **ManualValue:=_real_in_,** **Feedback:=_real_in_,** **InputPer:=_word_in_,** **FeedbackPer:=_word_in_,** **Disturbance:=_real_in_,** **ManualEnable:=_bool_in_,** **ManualUP:=_bool_in_,** **ManualDN:=_bool_in_,** **ActuatorH:=_bool_in_,** **ActuatorL:=_bool_in_,** **ErrorAck:=_bool_in_,** **Reset:=_bool_in_,** **ModeActivate:=_bool_in_,** **Mode:=_int_in_,** **ScaledInput=>_real_out_,** **ScaledFeedback=>_real_out_,** **ErrorBits=>_dword_out_,** **OutputPer=>_word_out_,** **State=>_int_out_,** **OutputUP=>_bool_out_,** **OutputDN=>_bool_out_,** **SetpoIntLimitH=>_bool_out_,** **SetpoIntLimitL=>_bool_out_,** **InputWarningH=>_bool_out_,** **InputWarningL=>_bool_out_,** **Error=>_bool_out_,** **ErrorBits=>_dword_out_);**	Инструкция PID_3Step конфигурирует ПИД-регулятор с функциями автонастройки, оптимизированный для моторизованных вентилей и исполнительных устройств. Он предлагает два булевых выхода. PID_3Step - это PIDT1 регулятор с анти-насыщением и оценкой П- и Д-составляющей.

[1] При добавлении инструкции, STEP 7 автоматически создает TO и DB экземпляра. DB экземпляра содержит параметры регулятора TO.

[2] В примере SCL "PID_3Step_1" - это имя DB экземпляра.

Таблица 10-23 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
Setpoint	IN	Real	Заданное значение ПИД-регулятора в автоматическом режиме (значение по умолчанию: 0,0)
Input	IN	Real	Переменная (тег) программы пользователя используется в качестве источника технологического значения. (значение по умолчанию: 0,0) При использовании параметра Input, требуется установка Config.InputPerOn = FALSE.
Input_PER	IN	Word	Аналоговый вход используется в качестве источника технологического значения. (значение по умолчанию: W#16#0) При использовании параметра Input_PER, требуется установка Config.InputPerOn = TRUE.
Actuator_H	IN	Bool	Цифровая обратная связь по положению вентиля для верхнего ограничителя Если Actuator_H = TRUE, значит вентиль находится на верхнем упоре и его движение далее в этом направлении не выполняется. (значение по умолчанию: FALSE)
Actuator_L	IN	Bool	Цифровая обратная связь по положению вентиля для нижнего ограничителя Если Actuator_L = TRUE, значит вентиль находится на нижнем упоре и его движение далее в этом направлении не выполняется. (значение по умолчанию: FALSE)
Feedback	IN	Real	Возвращенная позиция вентиля. (значение по умолчанию: 0,0) При использовании параметра Feedback, требуется установка Config.FeedbackPerOn = FALSE.
Feedback_PER	IN	Int	Возвращенное аналоговое значение позиции вентиля. (значение по умолчанию: W#16#0) При использовании параметра Feedback_PER, требуется установка Config.FeedbackPerOn = TRUE.. Feedback_PER масштабируется на основе следующих переменных (тегов): • Config.FeedbackScaling.LowerPointIn • Config.FeedbackScaling.UpperPointIn • Config.FeedbackScaling.LowerPointOut • Config.FeedbackScaling.UpperPointOut
Disturbance	IN	Real	Возмущающее воздействие или значение предварительного регулирования

Параметр и тип данных		Тип данных	Описание
ManualEnable	IN	Bool	Включает или выключает ручной режим. (значение по умолчанию: FALSE): • Изменение фронта с FALSE на TRUE активирует "Ручной режим работы", в то время как в состоянии = 4 режим работы не изменяется. Пока ManualEnable = TRUE, режим работы не может переключаться по положительному фронту на ModeActivate, также нельзя использовать диалоговое окно для ввода в эксплуатацию. • Изменение фронта с TRUE на FALSE активирует назначенный Mode режим работы. Примечание: Изменять режим работы рекомендуется через ModeActivate.
ManualValue	IN	Real	Технологическое значение для ручного режима. (значение по умолчанию: 0,0) В ручном режиме работы задается абсолютная позиция вентиля. ManualValue обрабатывается лишь в том случае, если используется OutputPer **или** если доступна возвращенная позиция.
ManualUP	IN	Bool	• Manual_UP = TRUE: – Вентиль открывается и в том случае, если используется Output_PER или обратная связь по положению. При достижении верхней конечной точки дальнейшее движение вентиля прекращается. – См. также Config.VirtualActuatorLimit • Manual_UP = FALSE: – Если используется Output_PER или обратная связь по положению, вентиль перемещается до ManualValue. В ином случае вентиль дальше не перемещается. Примечание: Если Manual_UP и Manual_DN одновременно устанавливаются на TRUE, то вентиль не перемещается.
ManualDN	IN	Bool	• Manual_DN = TRUE: – Вентиль открывается и в том случае, если используется Output_PER или обратная связь по положению. При достижении верхней конечной точки дальнейшее движение вентиля прекращается. – См. также Config.VirtualActuatorLimit • Manual_DN = FALSE: – Если используется Output_PER или обратная связь по положению, вентиль перемещается до ManualValue. В ином случае вентиль дальше не перемещается.
ErrorAck	IN	Bool	Сбрасывает ErrorBits и выход предупреждения. Изменение фронта с FALSE на TRUE

Параметр и тип данных		Тип данных	Описание
Reset	IN	Bool	Перезапускает регулятор. (значение по умолчанию: FALSE): • Изменение фронта с FALSE на TRUE: – Выполняет переключение в "неактивный" режим – Сбрасывает ErrorBits и выходы предупреждения. – Удаляет интегральную составляющую – Сохраняет параметры ПИД • Пока Reset = TRUE, PID_3Step остается в "неактивном" режиме (состояние = 0). • Изменение фронта с TRUE на FALSE: – PID_3Step переходит в режим работы, сохраненный в параметре Mode.
ModeActivate	IN	Bool	PID_3Step переключается в рабочее состояние, сохраненное в параметре Mode. Изменение фронта с FALSE на TRUE:
Mode	IN	Int	Требуемый режим работы ПИД; будет активироваться положительным фронтом на входе Mode Activate .
ScaledInput	OUT	Real	Масштабированное технологическое значение
ScaledFeedback	OUT	Real	Масштабированная обратная связь по положению вентиля Примечание: Для исполнительного устройства без обратной связи по положению указанная с помощью ScaledFeedback позиция исполнительного устройства является очень неточной. ScaledFeedback в этом случае может использоваться только для грубой оценки актуальной позиции.
Output_UP	OUT	Bool	Цифровое выходное значение для открытия вентиля. (значение по умолчанию: FALSE) Если Config.OutputPerOn = FALSE, то используется параметр Output_UP.
Output_DN	OUT	Bool	Цифровое выходное значение для закрытия вентиля. (значение по умолчанию: FALSE) Если Config.OutputPerOn = FALSE, то используется параметр Output_DN.
Output_PER	OUT	Word	Аналоговое выходное значение. Если Config.OutputPerOn = TRUE, то используется параметр Output_PER.
SetpointLimitH	OUT	Bool	Верхнее предельное заданное значение. (значение по умолчанию: FALSE) Если SetpointLimitH = TRUE, то абсолютное верхнее предельное значение заданного значения достигнуто (Setpoint ≥ Config.SetpointUpperLimit). Примечание: Заданное значение ограничено до (Setpoint ≥ Config.SetpointUpperLimit).
SetpointLimitL	OUT	Bool	Нижнее предельное заданное значение. (значение по умолчанию: FALSE) Если SetpointLimitL = TRUE, то абсолютное нижнее предельное значение заданного значения достигнуто (Setpoint ≥ Config.SetpointLowerLimit). Примечание: Заданное значение ограничено до (Setpoint ≥ Config.SetpointLowerLimit).

Параметр и тип данных		Тип данных	Описание
InputWarningH	OUT	Bool	Если InputWarningH = TRUE, то входное значение достигло или превысило верхнюю границу предупреждения. (значение по умолчанию: FALSE)
InputWarningL	OUT	Bool	Если InputWarningL = TRUE, то входное значение достигло или нарушило нижнюю границу предупреждения. (значение по умолчанию: FALSE)
State	OUT	Int	Текущий режим работы ПИД-регулятора. (значение по умолчанию: 0) Изменить режим работы можно при помощи входного параметра Mode и положительного фронта на ModeActivate:: • State = 0: Неактивно • State = 1: Предварительная настройка • State = 2: Ручная точная регулировка • State = 3: Автоматический режим • State = 4: Ручной режим • State = 5: Замещающее выходное значение • State = 6: Измерение времени перехода • State = 7: Контроль ошибок • State = 8: Замещающее выходное значение с контролем ошибок • State = 10: Ручной режим без сигналов ограничения
Error	OUT	Bool	Если Error = TRUE, то активно как минимум одно сообщение об ошибке. (значение по умолчанию: FALSE) Примечание: Параметр Error в V1.x ПИД было полем ErrorBits с кодами ошибок. Теперь булевский флаг является указанием на то, что возникла ошибка.
ErrorBits	OUT	DWord	В таблице параметров (Страница 610) ErrorBits для инструкции PID_3Step описываются активные сообщения об ошибках. (значение по умолчанию: DW#16#0000 (ошибки отсутствуют)). Параметр ErrorBits является сохраняющимся или сбрасывается по положительному фронту на Reset или ErrorAck. Примечание: В V1.x параметр ErrorBits был определен как параметр Error и отсутствовал.

Принцип работы регулятора PID_3Step

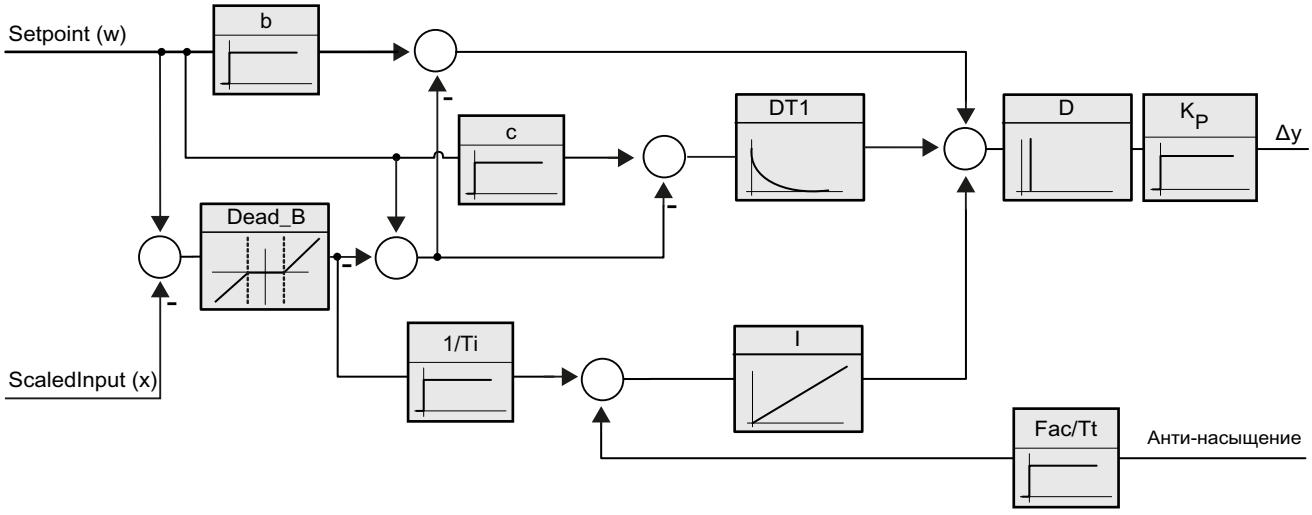


Рисунок 10-3 Принцип работы регулятора PID_3Step как PIDT1 регулятора с анти-насыщением

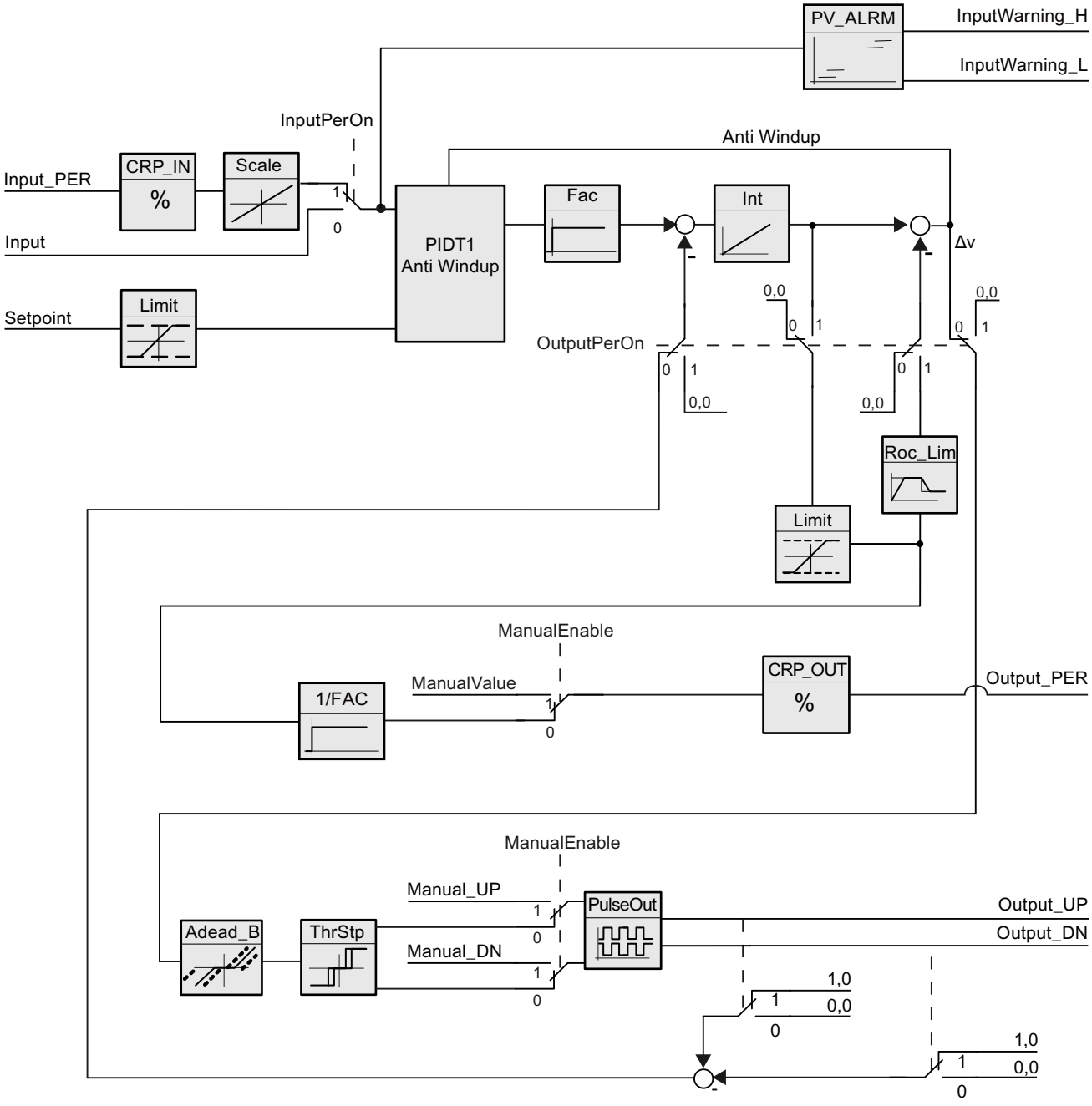


Рисунок 10-4 Принцип работы регулятора PID_3Step без обратной связи по положению

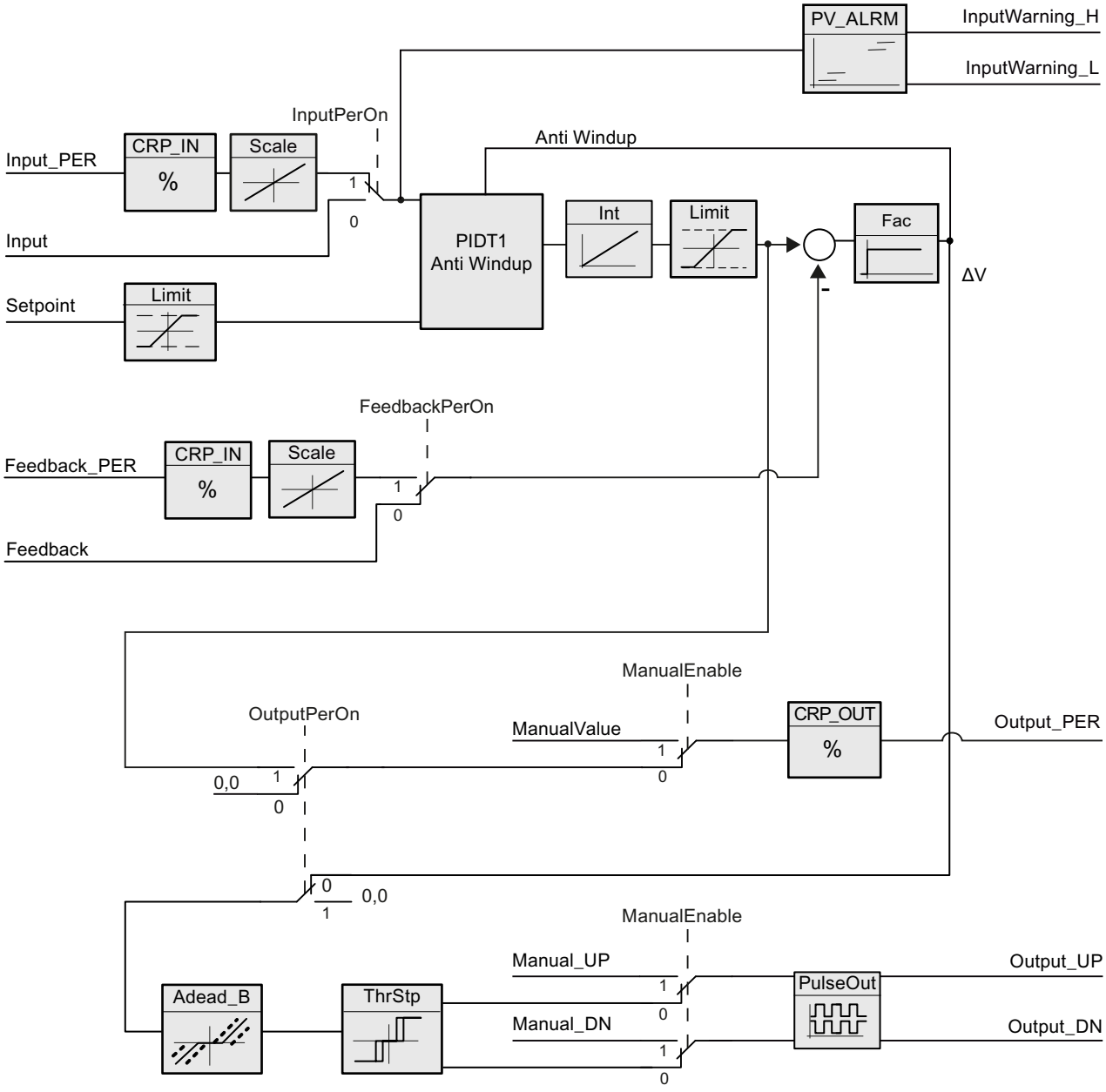


Рисунок 10-5 Принцип работы регулятора PID_3Step с активированной обратной связью по положению

10.2.3.2 Параметр ErrorBit инструкции PID_3Step

Если активно несколько ошибок, то значения кодов ошибок отображаются с помощью двоичного сложения. Например, отображение кода ошибки 0003 указывает на то, что также активны ошибки 0001 и 0002.

Таблица 10-24 Параметр ErrorBit инструкции PID_3STEP

ErrorBit (DW#16#...)	Описание
0000	Ошибки отсутствуют
0001 [1, 2]	Параметр Input находится вне допустимого диапазона технологических значений. Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 [2, 3]	Недействительное значение в параметре Input_PER. Проверить наличие ошибок на аналоговом входе.
0004 [4]	Ошибка при точной регулировке. Колебания технологического значения нельзя скомпенсировать.
0010 [4]	Заданное значение было изменено во время выполнения настройки. Примечание: Допустимые колебания на заданном значении могут быть установлены в переменной CancelTuningLevel.
0020	Предварительная настройка при точной регулировке не разрешена. Примечание: Если до возникновений ошибки имело место ActivateRecoverMode = TRUE, то PID_3Step остается в режиме точной регулировки.
0080 [4]	Ошибка при предварительной настройке. Неправильная конфигурация границ выходного значения. Проверить, правильно ли сконфигурированы границы выходного значения и соответствуют ли они логике регулирования.
0100 [4]	Ошибка при точной регулировке привела к недопустимым параметрам.
0200 [2, 3]	Недействительное значение в параметре Input: Числовой формат значения недействителен.
0400 [2, 3]	Расчет выходного значения не удался. Проверить параметры ПИД.
0800 [1, 2]	Ошибка времени выборки: PID_3Step не вызывается при активном времени выборки OB циклического прерывания.
1000 [2, 3]	Недействительное значение в параметре Setpoint: Числовой формат значения недействителен.
2000 [1, 2, 5]	Недействительное значение в параметре Feedback_PER. Проверить наличие ошибок на аналоговом входе.
4000 [1, 2, 5]	Недействительное значение в параметре Feedback: Числовой формат значения недействителен.

ErrorBit (DW#16#...)	Описание
8000 [1, 2]	Ошибка в цифровой обратной связи по положению. Actuator_H = TRUE и Actuator_L = TRUE. Исполнительный элемент не может перейти на замещающее выходное значение и остается на текущей позиции. В этом состоянии ручной режим невозможен. Для перемещения исполнительного элемента из этого состояния, необходимо отключить "Концевой ограничитель исполнительного элемента" (Config.ActuatorEndStopOn = FALSE) или перейти в ручной режим без сигналов ограничения (Mode = 10).
10000	Недействительное значение в параметре ManualValue: Числовой формат значения недействителен. Исполнительный элемент не может перейти на ручное значение и остается на текущей позиции. Задать действительное значение в ManualValue или переместить исполнительный элемент в ручном режиме при помощи Manual_UP и Manual_DN.
20000	Недопустимое значение переменной (тега) SavePosition: Числовой формат значения недействителен. Исполнительный элемент не может перейти на замещающее выходное значение и остается на текущей позиции.
40000	Недействительное значение в параметре Disturbance: Числовой формат значения недействителен. Примечание: Если до ошибки был активен автоматический режим и ActivateRecoverMode = FALSE, то Disturbance устанавливается на ноль. PID_3Step остается в автоматическом режиме. Примечание: Если до возникновения ошибки была активна предварительная настройка или точная регулировка и имело место ActivateRecoverMode = TRUE, то PID_3Step переходит в сохраненный в параметре Mode режим работы. Если ошибки на текущем этапе не влияют на выходное значение, то настройка не отменяется. Ошибка не действует при измерении времени перехода.

1 Примечание: Если до ошибки был активен автоматический режим и ActivateRecoverMode = TRUE, то PID_3Step остается в автоматическом режиме.

2 Примечание: Если до возникновения ошибки была активна предварительная настройка, точная регулировка или измерение времени перехода и имело место ActivateRecoverMode = TRUE, то PID_3Step переходит в сохраненный в параметре Mode режим работы.

3 Примечание: Если до ошибки был активен автоматический режим и ActivateRecoverMode = TRUE, то PID_3Step переходит в режим работы "Замещающее выходное значение с контролем ошибок" или "Контроль ошибок". Когда ошибка будет устранена, PID_3Step снова переходит в автоматический режим.

4 Примечание: Если до возникновения ошибки имело место ActivateRecoverMode = TRUE, то PID_3Step отменяет настройку и переходит в сохраненный в параметре Mode режим работы.

5 Исполнительный элемент не может перейти на замещающее выходное значение и остается на текущей позиции. Изменение позиции исполнительного элемента в ручном режиме возможно только при помощи Manual_UP и Manual_DN, но не ManualValue.

10.2.3.3 Параметры предупреждения инструкции PID_3Step

Если на ПИД-регуляторе активно несколько предупреждений, то значения кодов ошибок отображаются с помощью двоичного сложения. Например, отображение кода ошибки 0003 указывает на то, что также активны ошибки 0001 и 0002.

Таблица 10-25 Параметр Warning инструкции PID_Compact

Warning (DW#16#...)	Описание
0000	Предупреждения отсутствуют.
0001 [1]	Точка перегиба не была найдена во время предварительной настройки.
0002	При настройке в рабочем режиме были вызваны колебания. (Параметр "Warning" блокирует это предупреждение, оно доступно только в параметре “WarningInternal” для целей диагностики.)
0004 [1]	Заданное значение было ограничено до сконфигурированных пределов.
0008 [1]	Для выбранного методы вычисления не были определены все необходимые характеристики регулятора. В качестве замены параметры ПИД были рассчитаны при помощи метода TIR.TuneRuleHeat / TIR.TuneRuleCool = 3.
0010	Не удалось изменить рабочее состояние из-за Reset = TRUE или ManualEnable = TRUE.
0020	Время цикла вызывающего OB ограничивает время выборки ПИД алгоритма. Сокращение времени цикла OB может улучшить результаты.
0040 [1]	Технологическое значение превысило одну из границ предупреждения.
0080	Недопустимое значение в "Mode". Смена рабочего состояния выполнена не будет.
0100 [1]	Ручное значение было ограничено до предельных значений выхода регулятора.
0200	Указанное правило для настройки не поддерживается. Расчет параметров ПИД не выполняется.
1000	Замещающее выходное значение не может быть достигнуто, поскольку оно не соответствуют установленным для выходного значения пределам.

[1] Примечание: ПИД-регулятор, как только причина была устранена или реакция пользователя повторилась с действительными параметрами, автоматически удаляет следующие предупреждения: 0001, 0004, 0008, 0040 и 0100.

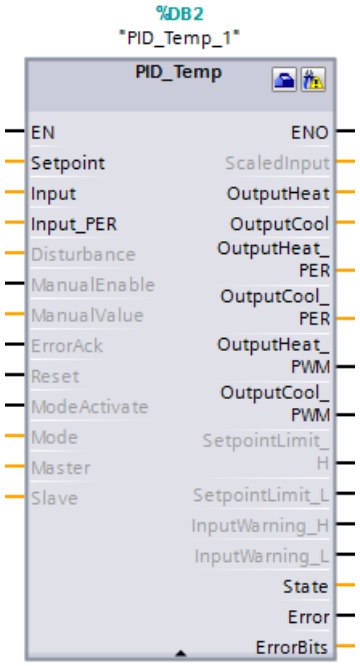
10.2.4 PID_Temp

10.2.4.1 Инструкция PID_Temp

Инструкция PID_Temp предоставляет универсальный ПИД-регулятор для специальных требований при регулировании температуры.

Таблица 10-26 Инструкция PID_Temp

LAD/FBD	SCL	Описание

	`"PID_Temp_1"(`
	`  Setpoint:=_real_in_,`
	`  Input:=_real_in_,`
	`  Input_PER:=_int_in_,`
	`  Disturbance:=_real_in_,`
	`  ManualEnable:=_bool_in_,`
	`  ManualValue:=_real_in_,`
	`  ErrorAck:=_bool_in_,`
	`  Reset:=_bool_in_,`
	`  ModeActivate:=_bool_in_,`
	`  Mode:=_int_in_,`
	`  Master:=_dword_in`
	`  Save:=_dword_in`
	`  ScaledInput=>_real_out_,`
	`  OutputHeat=>_real_out_,`
	`  OutputCool=>_real_out_,`
	`  OutputHeat_PER=>_int_out_,`
	`  OutputCool_PER=>_int_out_,`
	`  OutputHeat_PWM=>_bool_out_,`
	`  OutputCool_PWM=>_bool_out_,`
	
	`SetpointLimit_H=>_bool_out_,`
	
	`SetpointLimit_L=>_bool_out_,`
	`  InputWarning_H=>_bool_out_,`
	`  InputWarning_L=>_bool_out_,`
	`  State=>_int_out_,`
	`  Error=>_bool_out_,`
	`  ErrorBits=>_dword_out_);`
		PID_Temp предлагает следующие возможности:
		• Нагрев и охлаждение в процессе с различными исполнительными устройствами
		• Встроенная автоматическая настройка для регулирования температуры
		• Схема каскадного включения регуляторов для нескольких температур в процессе, регулируемых одним исполнительным устройством

[1] При добавлении инструкции, STEP 7 автоматически создает TO и DB экземпляра. DB экземпляра содержит параметры регулятора TO.

[2] В примере SCL "PID_Temp_1" - это имя DB экземпляра.

Таблица 10-27 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
Setpoint	IN	Real	Заданное значение ПИД-регулятора в автоматическом режиме (значение по умолчанию: 0,0)
Input	IN	Real	Переменная (тег) программы пользователя используется в качестве источника технологического значения. (значение по умолчанию: 0,0) При использовании параметра Input, требуется установка Config.InputPerOn = FALSE.
Input_PER	IN	Int	Аналоговый вход используется в качестве источника технологического значения. (значение по умолчанию: 0) При использовании параметра Input_PER, требуется установка Config.InputPerOn = TRUE.
Disturbance	IN	Real	Возмущающее воздействие или значение предварительного регулирования
ManualEnable	IN	Bool	Включает или выключает ручной режим. (значение по умолчанию: FALSE): • Изменение фронта с FALSE на TRUE активирует "Ручной режим работы", в то время как в состоянии = 4 режим работы не изменяется. Пока ManualEnable = TRUE, режим работы не может переключаться по положительному фронту на ModeActivate, также нельзя использовать диалоговое окно для ввода в эксплуатацию. • Изменение фронта с TRUE на FALSE активирует назначенный Mode режим работы. Примечание: Изменять режим работы рекомендуется через ModeActivate.
ManualValue	IN	Real	Выходное значение для ручного режима. (значение по умолчанию: 0,0) Допустимые значения: от Config.OutputLowerLimit до Config.OutputUpperLimit.
ErrorAck	IN	Bool	Сбрасывает ErrorBits и выходы предупреждения при изменении фронта с FALSE на TRUE. (значение по умолчанию: FALSE)
Reset	IN	Bool	Перезапускает регулятор. (значение по умолчанию: FALSE): • Изменение фронта с FALSE на TRUE: – Выполняет переключение в "неактивный" режим – Сбрасывает ErrorBits и выходы предупреждения. – Удаляет интегральную составляющую – Сохраняет параметры ПИД • Пока Reset = TRUE, PID_Temp остается в "неактивном" режиме (состояние = 0). • Изменение фронта с TRUE на FALSE: – PID_Temp переходит в режим работы, сохраненный в параметре Mode.
ModeActivate	IN	Bool	PID_Temp переключается при изменении фронта с FALSE на TRUE в рабочее состояние, сохраненное в параметре Mode. (значение по умолчанию: FALSE)

Параметр и тип данных		Тип данных	Описание
Mode	IN/OUT	Int	Активируется положительным фронтом на входе Mode Activate. Выбор режима работы (значение по умолчанию: 0,0): • Mode = 0: Неактивно • Mode = 1: Предварительная настройка • Mode = 2: Точная регулировка • Mode = 3: Автоматический режим • Mode = 4: Ручной режим "Замещающее выходное значение с контролем ошибок" (State = 5). Не может быть активировано пользователем; это автоматическая реакция на ошибку.
Master	IN/OUT	DWord	Каскадное включение на ведущее устройство (условия анти-насыщение и настройки). (значение по умолчанию: DW#16#0000) • Биты 0 - 15: Не используются в инструкции PID_Temp • Биты 16 - 23: Счетчик предельных значений: Ведомое устройство увеличивает это значение при достижении предельного значения. Число ведомых устройств с ограничением обрабатывается для функции анти-насыщения (см. параметр Config.Cascade.AntiWindUpMode). • Бит 24: IsAutomatic: Этот бит устанавливается на "1", когда все ведомые устройства этого регулятора находятся в автоматическом режиме и обрабатываются для проверки условий настройки в каскаде. Этот бит совпадает с параметром AllSlaveAutomaticState. • Бит 25: "IsReplacement-Setpoint": Этот бит устанавливается на "1", когда для одного ведомого устройства этого регулятора активировано "замещающее заданное значение" и он обрабатывается для проверки условий настройки в каскаде. Инвертированное значение сохраняется в параметре NoSlaveReplacementSetpoint.
Slave	IN/OUT	DWord	
ScaledInput	OUT	Real	Масштабированное технологическое значение. (значение по умолчанию: 0,0)
OutputHeat[1]	OUT	Real	Выходное значение для нагрева в формате REAL. (значение по умолчанию: 0,0) Это выходное значение рассчитывается независимо от выбора выхода с параметром Config.Output.Heat.Select.
OutputCool[1]	OUT	Real	Выходное значение для охлаждения в формате REAL. (значение по умолчанию: 0,0) Это выходное значение рассчитывается независимо от выбора выхода с параметром Config.Output.Cool.Select.
OutputHeat_PER[1]	OUT	Int	Выходное значение для нагрева в периферийном формате (значение по умолчанию: 0) Это выходное значение вычисляется только при его выборе с помощью параметра Config.Output.Heat.Select = 2. Если оно не выбрано, то выход всегда "0".
OutputCool_PER[1]	OUT	Int	Выходное значение для охлаждения в периферийном формате (значение по умолчанию: 0) Это выходное значение вычисляется только при его выборе с помощью параметра Config.Output.Cool.Select = 2. Если оно не выбрано, то выход всегда "0".

Параметр и тип данных		Тип данных	Описание
OutputHeat_PWM[1]	OUT	Bool	Широтно-импульсная модуляция выходного значения для нагрева. (значение по умолчанию: FALSE) Это выходное значение вычисляется только при его выборе с помощью параметра Config.Output.Heat.Select = 1 (значение по умолчанию). Если оно не выбрано, то выход всегда FALSE.
OutputCool_PWM[1]	OUT	Bool	Широтно-импульсная модуляция выходного значения для охлаждения. (значение по умолчанию: FALSE) Это выходное значение вычисляется только при его выборе с помощью параметра Config.Output.Cool.Select = 1 (значение по умолчанию). Если оно не выбрано, то выход всегда FALSE.
SetpointLimit_H	OUT	Bool	Верхнее предельное заданное значение. (значение по умолчанию: FALSE) Если SetpointLimit_H = TRUE, то абсолютное верхнее предельное значение заданного значения достигнуто (Setpoint ≥ Config.SetpointUpperLimit). Заданное значение ограничено до Config.SetpointUpperLimit.
SetpointLimit_L	OUT	Bool	Нижнее предельное заданное значение. (значение по умолчанию: FALSE) Если SetpointLimit_L = TRUE, то абсолютное нижнее предельное значение заданного значения достигнуто (Setpoint ≥ Config.SetpointLowerLimit). Заданное значение ограничено до Config.SetpointLowerLimit.
InputWarning_H	OUT	Bool	Если InputWarning_H = TRUE, то технологическое значение достигло или превысило верхнюю границу предупреждения. (значение по умолчанию: FALSE)
InputWarning_L	OUT	Bool	Если InputWarning_L = TRUE, то технологическое значение достигло или перешло нижнюю границу предупреждения. (значение по умолчанию: FALSE)
State	OUT	Int	Текущий режим работы ПИД-регулятора. (значение по умолчанию: 0) Изменить режим работы можно при помощи входного параметра Mode и положительного фронта на ModeActivate: • State = 0: Неактивно • State = 1: Предварительная настройка • State = 2: Точная регулировка • State = 3: Автоматический режим • State = 4: Ручной режим • State = 5: Замещающее выходное значение с контролем ошибок
Error	OUT	Bool	Если Error = TRUE, значит в этом цикле активно как минимум одно сообщение об ошибке. (значение по умолчанию: FALSE) Примечание: Параметр Error в V1.x ПИД было полем ErrorBits с кодами ошибок. Теперь булевский флаг является указанием на то, что возникла ошибка.

Параметр и тип данных		Тип данных	Описание
ErrorBits	OUT	DWord	В таблице параметров ErrorBits (Страница 624) для инструкции PID_Temp описываются активные сообщения об ошибках. (значение по умолчанию: DW#16#0000 (ошибки отсутствуют)). Параметр ErrorBits является сохраняющимся или сбрасывается по положительному фронту на Reset или ErrorAck. Примечание: В V1.x параметр ErrorBits был определен как параметр Error и отсутствовал.
Warning	OUT	DWord	В таблице параметров Warning (Страница 626) для инструкции PID_Temp описываются активные, значимые для пользователя предупреждающие сообщения. (значение по умолчанию: DW#16#0000 (предупреждений нет)).
WarningInternal	OUT	DWord	В таблице параметров WarningInternal для инструкции PID_Temp описываются активные, внутренние предупреждающие сообщения (включает в себя все предупреждения). (значение по умолчанию: DW#16#0000 (внутренние предупреждения отсутствуют)).

[1] Выходы параметров Output, Output_PER и Output_PWM могут использоваться параллельно.

Принцип работы регулятора PID_Temp

Выбор регулирование для нагрева и/или охлаждения

Сначала необходимо выбрать, необходимо ли охлаждающее устройство в дополнение к выходу нагрева в параметре "ActivateCooling". После этого необходимо определить, будут ли использоваться два набора ПИД параметров (расширенный режим) или только один набор ПИД параметров с дополнительным коэффициентом нагрева / охлаждения в параметре "AdvancedCooling".

Использование CoolFactor

Если будет использоваться коэффициент нагрева / охлаждения, вы необходимо определить значение вручную. Значение определяется на основе технических данных приложения (отношение пропорционального усиления исполнительных элементов (например, отношение максимальной мощности нагрева и охлаждения исполнительных элементов)) и назначается параметру "CoolFactor". Коэффициент нагрева / охлаждения 2.0 означает, что нагревательное устройство в два раза эффективнее охлаждающего устройства. При использовании коэффициента охлаждения, PID_Temp рассчитывает выходной сигнал и, в зависимости от его знака, умножает выходной сигнал на коэффициент нагрева / охлаждения (когда знак отрицательный) или нет (когда знак положительный).

Использование двух наборов ПИД-параметров

Различные наборы ПИД-параметров для нагрева и охлаждения могут автоматически определяться при вводе в эксплуатацию. Следует ожидать лучшего качества регулирования по сравнению с коэффициентом нагрева / охлаждения, потому что при использовании двух наборов параметров, в дополнение к разным пропорциональным усилениям, можно учитывать и различное время задержки. Недостатком является то, что процесс настройки может занять больше времени. Если переключение параметров PID активировано (Config.AdvancedCooling = TRUE), регулятор PID_Temp определяет в "автоматическом режиме" (регулирование активно), необходим ли в настоящий момент нагрев или охлаждение, и использует наборы параметров ПИД для регулирования.

ControlZone

С помощью регулятора PID_Temp для каждого набора параметров в параметре "ControlZone" можно определить зону регулирования. Если погрешность регулирования (заданное значение - вход) находится в пределах зоны регулирования, PID_Temp использует ПИД-алгоритм для расчета выходных сигналов. Однако, если погрешность регулирования выходит за пределы заданного диапазона, выход устанавливается на максимальное выходное значение нагрева или охлаждения (выход охлаждения активирован) / минимальное выходное значение нагрева (выход охлаждения деактивирован). Можно использовать эту функцию, чтобы быстрее достичь желаемого заданного значения, особенно для начального нагрева медленных температурных процессов.

DeadZone

В параметре "DeadZone" можно определить величину погрешности регулирования для нагрева и охлаждения, которая игнорируется ПИД-алгоритмом. Это означает, что погрешность регулирования в этом диапазоне подавляется, и регулятор PID_Temp ведет себя так, как если бы заданное значение и технологическое значения совпадали. Таким образом, можно сократить ненужные вмешательства со стороны регулятора в области заданного значения и сохранить исполнительный элемент. Для использования DeadZone необходимо определить значение вручную. Автонастройка не устанавливает значение DeadZone автоматически. DeadZone является симметричной (между -Retain.CtrlParams.Heat.DeadZone и + Retain.CtrlParams.Heat.DeadZone) для регуляторов нагрева без охлаждения или регуляторов нагрева / охлаждения с CoolFactor. DeadZone может быть асимметричной (между -Retain.CtrlParams.Cool.DeadZone и + Retain.CtrlParams.Heat.DeadZone) для регуляторов нагрева / охлаждения, использующих два набора параметров ПИД.

Принцип работы регуляторов PID_Temp

Следующие блок-схемы иллюстрируют стандартный и каскадный режим инструкций PID_Temp:

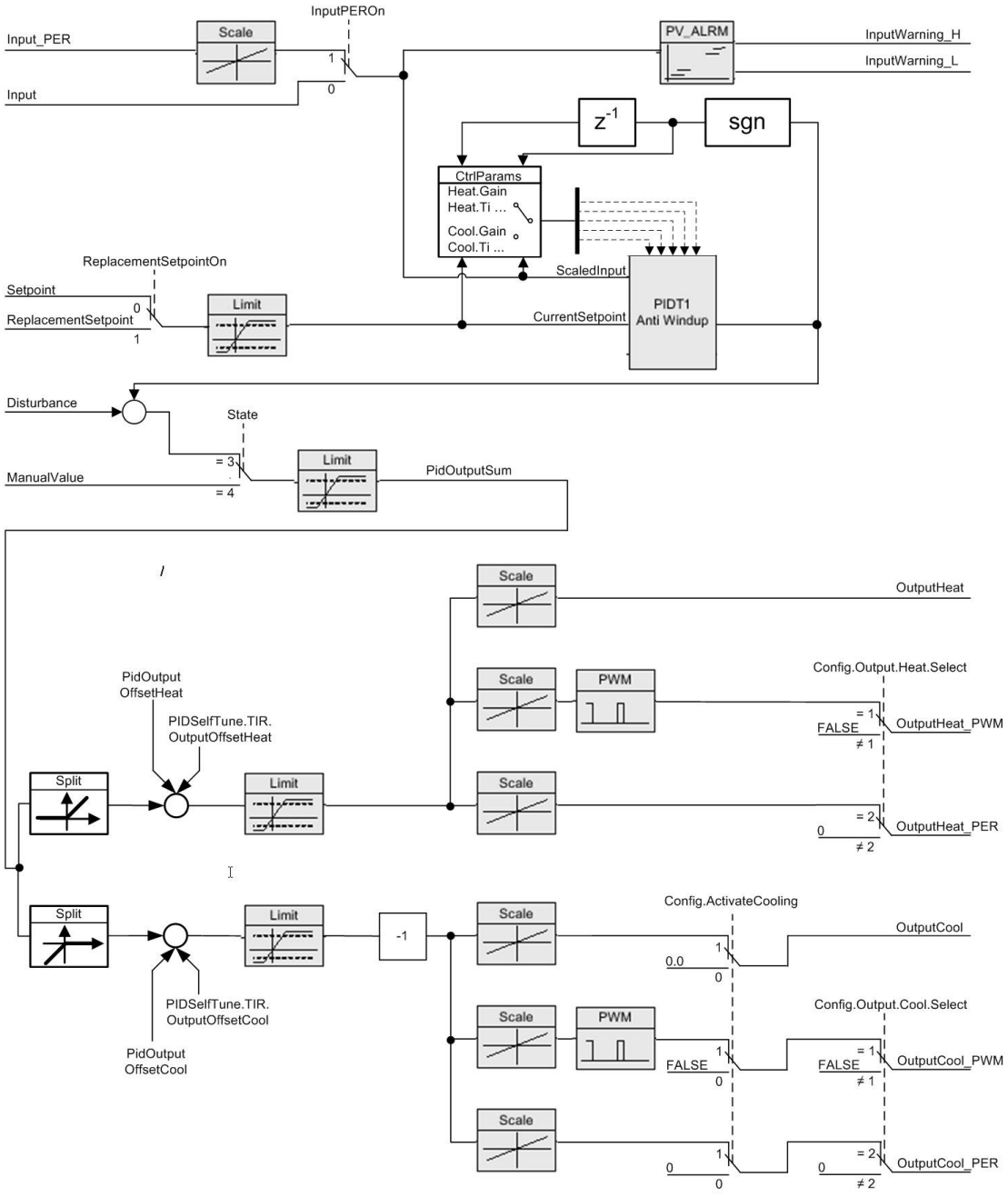


Рисунок 10-6 Блок-схема работы инструкции PID_Temp

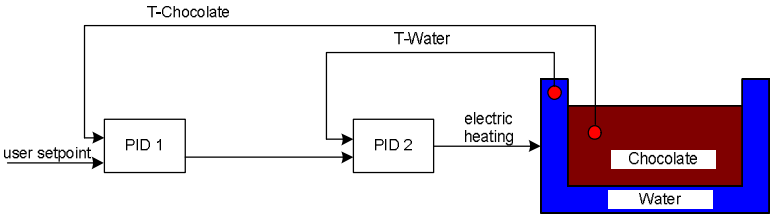


Рисунок 10-7 Блок-схема работы инструкции PID_Temp в каскадном режиме

Каскадирование регуляторов

Допускается каскадное включение ПИД-регуляторов температуры для обработки более одной температуры, регулируемой одним и тем же исполнительным элементом.

Последовательность вызова

Каскадные ПИД-регуляторы должны вызывать в одном и том же цикле OB. Сначала должно быть вызвано ведущее устройство, затем последующие ведомые устройства по потоку сигнала регулирования и, в заключении, последнее ведомое устройство в каскаде. Инструкция PID_Temp не выполняет автоматическую проверку порядка вызовов.

Коммуникационные соединения

При каскадировании регуляторов необходимо соединить ведущее и ведомое устройство таким образом, чтобы они могли обмениваться информацией друг с другом. Необходимо соединить параметр IN / OUT ведущего устройства с параметром IN / OUT ведомого устройства в направлении потока сигналов.

Дальше будет показано соединение регуляторов PID_Temp в каскаде с двумя подкаскадами: "PID_Temp1" предоставляет заданное значение. В конфигурации выходы "PID_Temp2", "PID_Temp3", "PID_Temp5", "PID_Temp6" и "PID_Temp8" соединены с процессом:

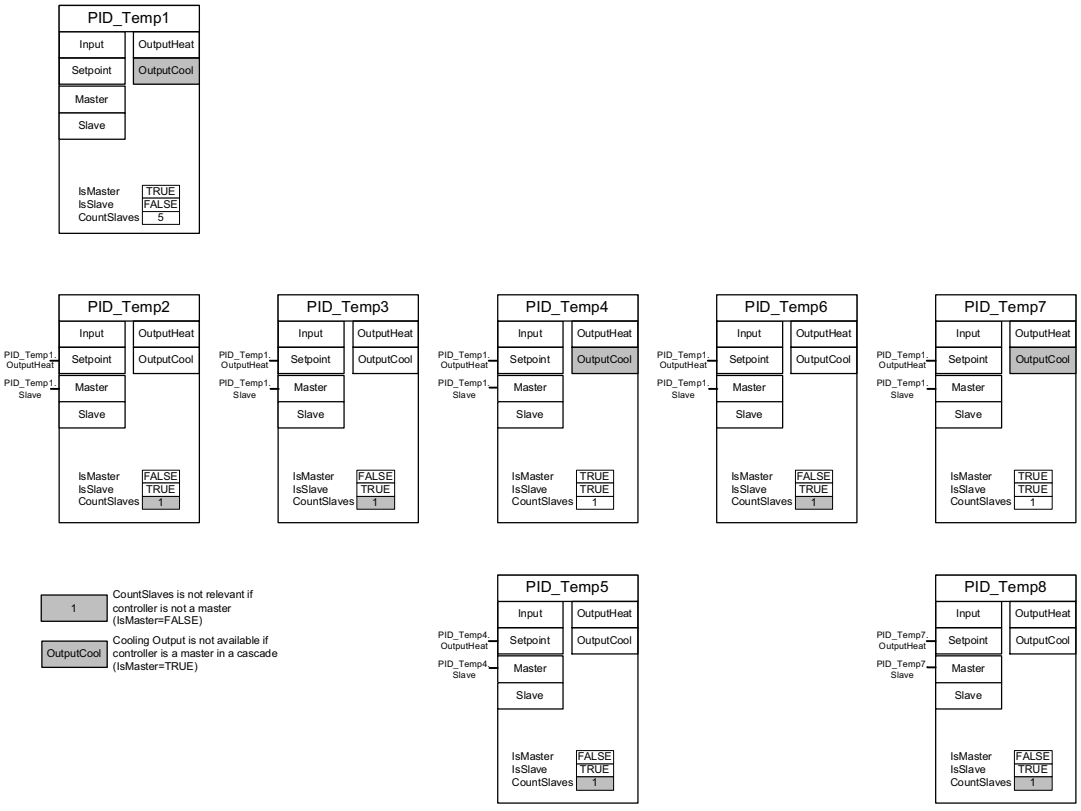


Рисунок 10-8 Коммуникационные соединения при каскадировании PID_Temp

Замещающее заданное значение

Инструкция PID_Temp предоставляет второй вход заданного значения в параметре "ReplacementSetpoint", который можно активировать, установив параметр "ReplacementSetpointOn" = TRUE. Можно использовать "ReplacementSetpoint" в качестве входа заданного значения при вводе в эксплуатацию или настройке ведомого регулятора, не разрывая соединение выхода и заданного значения между ведущим и ведомым устройством. Это соединение необходимо для нормальной работы каскада.

Таким образом, не нужно менять программу и загружать ее, если требуется временно отсоединить ведомое устройство от его ведущего устройства. Нужно лишь активировать "ReplacementSetpoint" и снова деактивировать его после. Значение воздействует на алгоритм ПИД, если можно увидеть заданное значение в параметре "CurrentSetpoint".

Автоматическая настройка

Автоматическая настройка для каскадного ведущего регулятора должна отвечать следующим требованиям:

- Ввод в эксплуатацию от внутреннего ведомого устройства к первому ведущему устройству.
- Все ведомые устройства ведущего устройства должны находиться в автоматическом режиме.
- Выход ведущего устройства должен быть заданным значением для ведомых устройство.

Инструкция PID_Temp оказывает следующую поддержку для автонастройки при каскадном включении:

- Когда автонастройка запускается для ведущего регулятора, он проверяет, все ли ведомые устройства находятся в автоматическом режиме и отключена ли функциональность замещающего заданного значения для всех ведомых устройств ("ReplacementSetpointOn" = FALSE). Если эти условия не выполняются, то автонастройка ведущего устройства невозможна. Ведущее устройство отменяет настройку, переходит в неактивный режим (если "ctivateRecoverMode" = FALSE) или обратно в режим, сохраненный в параметре "Режим" (если "ActivateRecoverMode" = TRUE). Ведущее устройство отображает сообщение об ошибке 200000hex ("Ошибка ведущего устройства в каскаде. Ведомые устройства не в автоматическом режиме или для них активировано замещающее заданное значение, что препятствует настройке ведущего устройства.").
- Если все ведомые устройства находятся в автоматическом режиме, система устанавливает параметр "AllSlaveAutomaticState" = TRUE. Можно использовать этот параметр в своих программах или локализовать причину ошибки 200000hex.
- Когда "ReplacementSetpoint" деактивируется для всех ведомых устройств, система устанавливает параметр "NoSlaveReplacementSetpoint" = TRUE. Можно использовать этот параметр в своих программах или локализовать причину ошибки 200000hex.

При использовании диалогового окна ввода в эксплуатацию инструкции PID_Temp, там можно найти дополнительную поддержку при настройке каскадного включения (Страница 646).

Режимы работы и обработка ошибок

Регулятор PID_Temp не позволяет переключать режим работы своему ведущему или ведомым устройствам. Это означает, что ведущее устройство внутри каскада остается в своем текущем режиме работы, когда ведомое устройство сигнализирует ошибку. Это является преимуществом, когда два или более параллельных ведомых устройств работают с этим главным регулятором: ошибка в одной цепочке не ведет к отключению параллельной цепочки.

Аналогичным образом, ведомое устройство внутри каскада остается в своем текущем режиме работы, если на его ведущем устройстве происходит ошибка. Однако дальнейшая работа ведомого устройства зависит от конфигурации ведущего устройства, поскольку заданное значение ведомого устройства является выходом ведущего устройства. Это означает, что если в конфигурации ведущего устройства с "ActivateRecoverMode" = TRUE возникает ошибка, то мастер выводит последнее действительное или замещающее выходное значение в качестве заданного значения

для ведомого устройства. Если для ведущего устройства конфигурируется "ActivateRecoverMode" = FALSE, то ведущее устройство переключается в неактивный режим и устанавливает все выходы на "0.0", так что ведомое устройство использует "0.0" в качестве заданного значения.

Поскольку только ведомые регуляторы имеют прямой доступ к исполнительным элементам, и они остаются в своем рабочем режиме в случае ошибки ведущего устройства, это позволяет избежать нарушения процесса. Например, для устройств по переработке пластмасс критической является ситуация, когда ведомые устройства прекращают работу и исполнительные механизмы отключаются, вследствие чего пластик может затвердеть внутри устройства, и всё это только из-за ошибки главного регулятора.

Анти-насыщение

Ведомое устройство в каскаде получает свое заданное значение с выхода ведущего устройства. Если ведомое устройство достигает своих собственных предельных значений для выхода, в то время как ведущее устройство все еще определяет погрешность регулирования (заданное значение - вход), ведущее устройство замораживается или уменьшает свой интеграционный вклад, чтобы предотвратить так называемое "интегральное насыщение". В случае "интегрального насыщения" ведущее устройство увеличивает свой интеграционный вклад до очень большого значения и должно сначала уменьшить его, прежде чем регулятор снова сможет нормально реагировать. Такое "интегральное насыщение" отрицательно влияет на динамику регулирования. Регулятор PID_Temp предлагает пути для предотвращения этого эффект в каскаде. Для этого конфигурируется параметр "Config.Cascade.AntiWindUpMode" главного регулятора:

Значение	Описание
0	Выключение функции анти-насыщения.
1	Уменьшение интеграционного вклада главного регулятора в отношении "ведомые устройства в ограничении" к "существующим ведомым устройствам" (параметр "CountSlaves").
2	Замораживание интеграционного вклада ведущего устройства при достижении ведомым устройством своего предельного значения. Значимо, только если "Config.Cascade.IsMaster" = TRUE.

10.2.4.2 Параметр ErrorBit из PID_Temp

Если на ПИД-регуляторе активно несколько предупреждений, то значения кодов ошибок отображаются с помощью двоичного сложения. Например, отображение кода ошибки 0003 указывает на то, что также активны ошибки 0001 и 0002.

Таблица 10-28 Параметр ErrorBit инструкции PID_Temp

ErrorBit (DW#16#...)	Описание
0000	Ошибки отсутствуют
0001 [1, 2]	Параметр Input находится вне допустимого диапазона технологических значений. Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 [2, 3]	Недействительное значение в параметре Input_PER. Проверить наличие ошибок на аналоговом входе.
0004 [4]	Ошибка при точной регулировке. Колебания технологического значения нельзя скомпенсировать.
0008 [4]	Ошибка при запуске предварительной настройки. Технологическое значение слишком приближено к заданному значению. Запустить точную регулировку.
0010 [4]	Заданное значение было изменено во время выполнения настройки. Примечание: Допустимые колебания на заданном значении могут быть установлены в переменной CancelTuningLevel.
0020	Предварительная настройка при точной регулировке не разрешена. Примечание: Если до возникновений ошибки имело место ActivateRecoverMode = TRUE, то PID_Temp остается в режиме точной регулировки.
0040 [4]	Ошибка при предварительной настройке. Охлаждению не удалось уменьшить технологическое значение.
0080 [4]	Ошибка при предварительной настройке. Неправильная конфигурация границ выходного значения. Проверить, правильно ли сконфигурированы границы выходного значения и соответствуют ли они логике регулирования.
0100 [4]	Ошибка при точной регулировке привела к недопустимым параметрам.
0200 [2, 3]	Недействительное значение в параметре Input: Числовой формат значения недействителен.
0400 [2, 3]	Расчет выходного значения не удался. Проверить параметры ПИД.
0800 [1, 2]	Ошибка времени выборки: PID_Temp не вызывается при активном времени выборки OB циклического прерывания.
1000 [2, 3]	Недействительное значение в параметре Setpoint: Числовой формат значения недействителен.
10000	Недействительное значение в параметре ManualValue: Числовой формат значения недействителен. Примечание: Если до возникновений ошибки имело место ActivateRecoverMode = TRUE, то PID_Temp использует SubstituteOutput как выходное значение. Как только в параметре ManualValue будет задано действительное значение, PID_Temp использует его в качестве выходного значения.

ErrorBit (DW#16#...)	Описание
20000	Недопустимое значение переменной (тега) SubstituteValue: Числовой формат значения недействителен. PID_Temp использует нижнюю границу выходного значения в качестве выходного значения. Примечание: Если до ошибки был активен автоматический режим,ActivateRecoverMode = TRUE и ошибка больше неактивна, PID_Temp снова перейдет в автоматический режим.
40000	Недействительное значение в параметре Disturbance: Числовой формат значения недействителен. Примечание: Если до ошибки был активен автоматический режим и ActivateRecoverMode = FALSE, то Disturbance устанавливается на ноль. PID_Temp остается в автоматическом режиме. Примечание: Если до возникновения ошибки была активна предварительная настройка или точная регулировка и имело место ActivateRecoverMode = TRUE, то PID_Temp переходит в сохраненный в параметре Mode режим работы. Если ошибки на текущем этапе не влияют на выходное значение, то настройка не отменяется.
200000	Ошибка ведущего устройства в каскаде. Ведомые устройства не в автоматическом режиме или для них активировано замещающее заданное значение, что препятствует настройке ведущего устройства.
400000	ПИД-регулятор блокирует предварительную настройку для нагрева, пока активно охлаждение.
800000	Технологическое значение должно быть приближено к заданному значению, чтобы запустить предварительную настройку для охлаждения.
1000000	Ошибка при запуске точной регулировки. "Heat.EnableTuning" и "Cool.EnableTuning" не определены или не соответствуют конфигурации.
2000000	Условием предварительной настройки для охлаждения является успешная предварительная настройка для нагрева.
4000000	Ошибка при запуске точной регулировки. "Heat.EnableTuning" и "Cool.EnableTuning" не могут быть установлены одновременно.
8000000	Ошибка при расчете параметра ПИД привела к неверным параметрам (например, отрицательное усиление; текущие параметры ПИД остаются неизменными, а настройка не действует)

[1] Примечание: Если до ошибки был активен автоматический режим и ActivateRecoverMode = TRUE, то PID_Temp остается в автоматическом режиме.

[2] Примечание: Если до возникновения ошибки была активна предварительная настройка или точная регулировка и имело место ActivateRecoverMode = TRUE, то PID_Temp переходит в сохраненный в параметре Mode режим работы.

[3] Примечание: Если до ошибки был активен автоматический режим и ActivateRecoverMode = TRUE, PID_Compact возвращает сконфигурированное замещающее выходное значение. Когда ошибка будет устранена, PID_Temp снова переходит в автоматический режим.

[4] Примечание: Если до возникновения ошибки имело место ActivateRecoverMode = TRUE, то PID_Temp отменяет настройку и переходит в сохраненный в параметре Mode режим работы.

10.2.4.3 Параметры предупреждений PID_Temp

Если на ПИД-регуляторе активно несколько предупреждений, то значения кодов ошибок отображаются с помощью двоичного сложения. Например, отображение кода ошибки 0003 указывает на то, что также активны ошибки 0001 и 0002.

Таблица 10-29 Параметр Warning инструкции PID_Temp

Warning (DW#16#...)	Описание
0000	Предупреждения отсутствуют.
0001 [1]	Точка перегиба не была найдена во время предварительной настройки.
0002	При настройке в рабочем режиме были вызваны колебания. (Параметр "Warning" блокирует это предупреждение, оно доступно только в параметре “WarningInternal” для целей диагностики.)
0004 [1]	Заданное значение было ограничено до сконфигурированных пределов.
0008 [1]	Для выбранного методы вычисления не были определены все необходимые характеристики регулятора. В качестве замены параметры ПИД были рассчитаны при помощи метода TIR.TuneRuleHeat / TIR.TuneRuleCool = 3.
0010	Не удалось изменить рабочее состояние из-за Reset = TRUE или ManualEnable = TRUE.
0020	Время цикла вызывающего OB ограничивает время выборки ПИД алгоритма. Сокращение времени цикла OB может улучшить результаты.
0040 [1]	Технологическое значение превысило одну из границ предупреждения.
0080	Недопустимое значение в "Mode". Смена рабочего состояния выполнена не будет.
0100 [1]	Ручное значение было ограничено до предельных значений выхода регулятора.
0200	Указанное правило для настройки не поддерживается. Расчет параметров ПИД не выполняется.
1000	Замещающее выходное значение не может быть достигнуто, поскольку оно не соответствуют установленным для выходного значения пределам.
4000	Указанный выбор выхода для нагрева и/или охлаждение не поддерживается. Активны только OutputHeat и OutputCool.
8000	Указанное значение для параметра PIDSelfTune.SUT.AdaptDelayTime не поддерживается, поэтому используется значение по умолчанию "0".
10000	Указанное значение для параметра PIDSelfTune.SUT.CoolingMode не поддерживается, поэтому используется значение по умолчанию "0".

[1] Примечание: ПИД-регулятор, как только причина была устранена или реакция пользователя повторилась с действительными параметрами, автоматически удаляет следующие предупреждения: 0001, 0004, 0008, 0040 и 0100.

10.2.5 Конфигурирование регуляторов PID_Compact и PID_3Step

Параметры технологического объекта определяют принцип работы ПИД-регулятора. Кликнуть по значку, чтобы открыть редактор конфигурации.

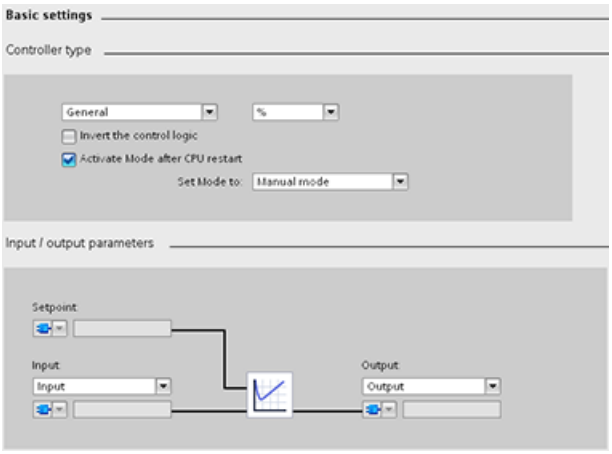


Таблица 10-30 Примеры настроек конфигурации для инструкции PID_Compact

Настройки		**Описание**
Основы	Тип регулятора	Выбирает технические единицы.
	Инверсия логики регулировки	Обеспечивает выбор ПИД-регулятора обратного действия. • Если эта опция не выбрана, то ПИД-регулятор находится в режиме прямого действия, и выход ПИД-регулятора растет, если входное значение < заданного значения. • Если эта опция выбрана, то выход контура ПИД-регулирования растет, если входное значение > заданного значения.
	Активация последнего режима работы после перезапуска CPU	Перезапускает ПИД-регулятор после сброса или превышения предела входа и возврата в допустимый диапазон.
	Вход	Выбирает параметр Input или параметр Input_PER (для аналогового) для технологического значения. Input_PER может поступать непосредственно из модуля аналогового входов.
	Выход	Выбирает параметр Output или параметр Output_PER (для аналогового) для выходного значения. Output_PER может поступать непосредственно на модуль аналоговых выходов.
Технологическое значение	Масштабирует как диапазон, так и пределы для технологического значения. При нарушении технологическим значением нижней или верхней границы, контур ПИД-регулирования переходит в неактивный режим работы и устанавливает выходное значение на 0. Для использования Input_PER, **необходимо** масштабировать аналогового технологическое значение (входное значение).	

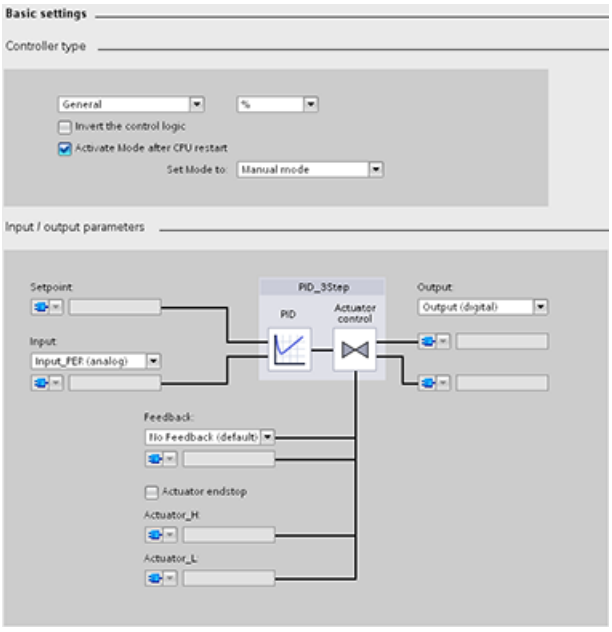


Таблица 10-31 Примеры настроек конфигурации для инструкции PID_3Step

Настройки		**Описание**
Основы	Тип регулятора	Выбирает технические единицы.
	Инверсия логики регулировки	Обеспечивает выбор ПИД-регулятора обратного действия. • Если эта опция не выбрана, то ПИД-регулятор находится в режиме прямого действия, и выход ПИД-регулятора растет, если входное значение < заданного значения. • Если эта опция выбрана, то выход контура ПИД-регулирования растет, если входное значение > заданного значения.
	Активация режима работы после перезапуска CPU	Перезапускает ПИД-регулятор после сброса или превышения предела входа и возврата в допустимый диапазон. Настройка режима работы: Определяет режим работы, в который должен перейти ПИД-регулятор после перезапуска.
	Вход	Выбирает параметр Input или параметр Input_PER (для аналогового) для технологического значения. Input_PER может поступать непосредственно из модуля аналогового входов.
	Выход	Указывает, должны ли для выходного значения использоваться цифровые выходы (Output_UP и Output_DN) или аналоговый выход (Output_PER).
	Подтверждение	Указывает тип состояния устройства, возвращаемого на ПИД-регулятор: • Без подтверждения (по умолчанию) • Подтверждение • Feedback_PER
Технологическое значение	Масштабирует как диапазон, так и пределы для технологического значения. При нарушении технологическим значением нижней или верхней границы, контур ПИД-регулирования переходит в неактивный режим работы и устанавливает выходное значение на 0. Для использования Input_PER, **необходимо** масштабировать аналогового технологическое значение (входное значение).	

Настройки		Описание
Исполнительный элемент	Время перехода двигателя	Устанавливает время от открытого до закрытого состояния вентиля. (Это значение можно найти в техническом паспорте или на шильдике вентиля.)
	Минимальное время включения	Устанавливает минимальное время срабатывания для вентиля. (Это значение можно найти в техническом паспорте или на шильдике вентиля.)
	Минимальное время выключения	Устанавливает минимальное время паузы для вентиля. (Это значение можно найти в техническом паспорте или на шильдике вентиля.)
	Реакция на ошибку	Определяет поведение вентиля при обнаружении ошибки или при сбросе ПИД-регулятора. Если будет решено использовать замещающую позицию, ввести "Безопасная позиция". Для аналогового подтверждения или аналогового выхода выбрать значение между верхним или нижним пределом для выхода. Для цифровых выходов можно выбрать только 0% (выкл) или 100% (вкл).
	Масштабирование обратной связи по положению[1]	• "Верхний концевой ограничитель" и "Нижний концевой ограничитель" определяют максимальную положительную позицию (полностью открытое положение) и максимальную отрицательную позицию (полностью закрытое положение). "Верхний концевой ограничитель" должен быть больше "Нижнего концевого ограничителя" • "Верхнее предельное технологическое значение" и "Нижнее предельное технологическое значение" определяют верхнее и нижнее положения вентиля во время настройки и в автоматическом режиме. • "FeedbackPER" ("Низкий" и "Высокий") определяет аналоговую обратную связь по положению вентиля. "FeedbackPER High" должно быть больше, чем "FeedbackPER Low".
Расширенные настройки	Контроль технологического значения	Устанавливает верхнее и нижнее предельное значение для технологического значения
	Параметры ПИД	В этом окне пользователь при необходимости может ввести собственные параметры настройки ПИД. Для этого должен быть установлен флажок "Активировать ручной ввод".

[1] "Масштабирование обратной связи по положению" может быть изменено только после активации "Подтверждения" в базовых настройках.

10.2.6 Конфигурирование регулятора PID_Temp

Параметры технологического объекта определяют принцип работы ПИД-регулятора. Кликнуть по значку, чтобы открыть редактор конфигурации.

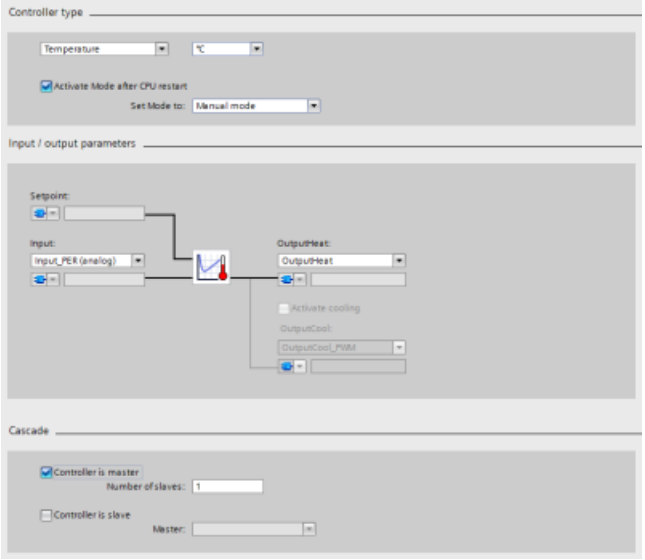


Таблица 10-32 Примеры настроек конфигурации для инструкции PID_Temp

Настройки		Описание
Основы	Тип регулятора	Выбирает технические единицы.
	Активация режима работы после перезапуска CPU	Перезапускает ПИД-регулятор после сброса или превышения предела входа и возврата в допустимый диапазон. Настройка режима работы: Определяет режим работы, в который должен перейти ПИД-регулятор после перезапуска.
	Вход	Выбирает параметр Input или параметр Input_PER (для аналогового) для технологического значения. Input_PER может поступать непосредственно из модуля аналогового входов.
	Выход нагрева	Указывает, должны ли для выходного значения использоваться цифровые выходы (OutputHeat и OutputHeat_PWM) или аналоговый выход (OutputHeat_PER (analog)).
	Выход охлаждения	Указывает, должны ли для выходного значения использоваться цифровые выходы (OutputCool и OutputCool_PWM) или аналоговый выход (OutputCool_PER (analog)).
Технологическое значение	Масштабирует как диапазон, так и пределы для технологического значения. При нарушении технологическим значением нижней или верхней границы, контур ПИД-регулирования переходит в неактивный режим работы и устанавливает выходное значение на 0. Для использования Input_PER, **необходимо** масштабировать аналогового технологическое значение (входное значение).	
Каскад	Регулятор это ведущее устройство	Установка регулятора в качестве ведущего устройства и выбор числа ведомых устройств.
	Регулятор это ведомое устройство	Установка регулятора в качестве ведомого устройства и выбор числа ведущих устройств.

Тип регулятора

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Физическая величина	"PhysicalQuantity"	Int (Enum)	• Общее • Температура (= по умолчанию)	Предварительный выбор для значения физической единицы Без многозначного регулирования и без возможности редактирования в онлайн-режиме функционального просмотра.
Единица измерения	"PhysicalUnit"	Int (Enum)	• Общее: Единицы = % • Температура: Единицы (возможный выбор) = – °C (= по умолчанию) – °F – K	Пользовательский выбор единиц снова сбрасывается на “0” при изменении физической величины.

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Активация режима работы после перезапуска CPU	"RunModeByStartup"	Bool	Флажок	Если установлено на TRUE (= по умолчанию), то регулятор переключается в состояние, сохраненное в переменной “Mode”, после цикла включения и выключения питания (Питание вкл - выкл - вкл) или перехода PLC из STOP в RUN. В остальных случаях PID_Temp остается в режиме “неактивный”.
Настройка режима работы	"Mode"	Int (Enum)	Режимы работы (возможный выбор): • 0: Неактивно • 1: Предварительная настройка • 2: Точная регулировка • 3: Автоматический режим • 4: Ручной режим (= по умолчанию):	Инженерная станция (ES) устанавливает стартовое значение переменной “Mode” согласно выбору пользователя. Значением по умолчанию для Mode (сохранено в TO-DB) является ручной режим.

Входные / выходные параметры

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Заданное значение	Заданное значение	Real	Real	Доступно только на странице свойств. Без многозначного регулирования в онлайн-режиме функционального просмотра.
Выбор входа	"Config.InputPerOn"	BOOL (Enum)	Bool	Выбор используемого входа. Возможный выбор: • FALSE: "Input" (Real) • TRUE: "Input_PER (аналоговый)"

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Вход	Вход или Input_PER	Real или Int	Real или Int	Доступно только на странице свойств. Без многозначного регулирования в онлайн-режиме функционального просмотра.
Выбор выхода (нагрев)	"Config.Output.Heat.Select"	Int (Enum)	2 >= Config.Output.Heat.Select >= 0	Выбор используемого для нагрева выхода. Возможный выбор: • "OutputHeat" (Real) • "OutputHeat_PWM" (Bool) (= по умолчанию) • "OutputHeat_PER (аналоговый)" (Word) Однократно устанавливается на "OutputHeat"Ю\, если пользователем в разделе "Каскад" установлен флажок "Это регулятор является ведущим устройством".
Выход (нагрев)	OutputHeat, OutputHeat_PER или OutputHeat_PWM	Real или Int или Bool	Real или Int или Bool	Доступно только на странице свойств. Без многозначного регулирования в онлайн-режиме функционального просмотра.

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Активировать выход (охлаждение)	"Config.ActivateCooling"	Bool	Bool	Если этот флажок будет установлен: • однократная установка "Config.Output.Heat.PidLowerLimit = 0.0. • Если параметр "Config.ActivateCooling" устанавливается на TRUE, вместо FALSE, если флажок не установлен (= по умолчанию). • Активируются все другие регуляторы "Выход (охлаждение)" (в "Основных настройках" и других представлениях). • Цвет линии от символа ПИД к регуляторам изменяется с серого на черный. • В разделе "Каскад" снимается флажок "Это регулятор является ведущим устройством". Примечание: Доступно, только если регулятор не конфигурируется как ведущее устройство для каскада (в разделе "Каскад" флажок "Это регулятор является ведущим устройством" снят); "Config.Cascade.IsMaster" = FALSE).

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Выбор выхода (охлаждение)	"Config.Output.Cool.Select"	Int (Enum)	2 >= Config.Output.Heat.Select >= 0	Выбор используемого для охлаждения выхода. Возможный выбор: • "OutputCool" (Real) • "OutputCool_PWM" (Bool) (= по умолчанию) • "OutputCool_PER (аналоговый)" (Word) Доступно, только если активируется "Активировать выход (охлаждение)"; (Config.ActivateCooling = TRUE).
Выход (охлаждение)	OutputCool, OutputCool_PER или OutputCool_PWM	Real или Int или Bool	Real или Int или Bool	Доступно только на странице свойств. Без многозначного регулирования в онлайн-режиме функционального просмотра.

Параметры каскада

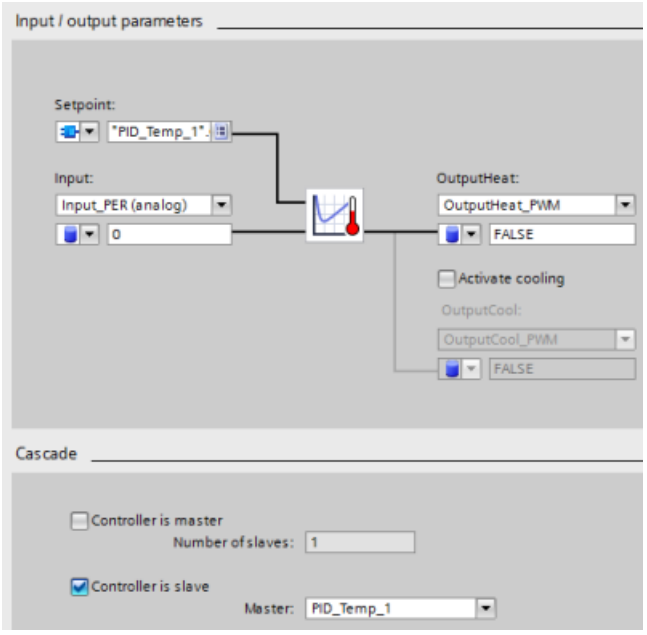
Следующие параметры позволяют выбирать регуляторы в качестве ведущих или ведомых устройств и определять число ведомых регуляторов, которые получают свое заданное значение непосредственно от главного регулятора:

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Это регулятор является ведущим устройством	"Config.Cascade.IsMaster"	Bool	Bool	Показывает, является ли этот регулятор ведущим устройством в каскаде. Установка флажка выполняет следующие действия: • Параметр "Config.Cascade.IsMaster" устанавливается на TRUE, вместо FALSE, если флажок не установлен (= по умолчанию). • Устанавливает "Выбор выхода (нагрев)" в разделе "Входные / выходные параметры" однократно на "OutputHeat" (Config.Output.Heat.Select = 0). • Входное поле "Количество ведомых устройств" активируется. • Флажок "Активировать выход (охлаждение)" в раздел "Входные / выходные параметры" снимается. Примечание: Доступно, только если выход охлаждения этого регулятора деактивирован (флажок "Активировать выход (охлаждение)" в разделе "Входные / выходные параметры" снят (Config.ActivateCooling = FALSE).

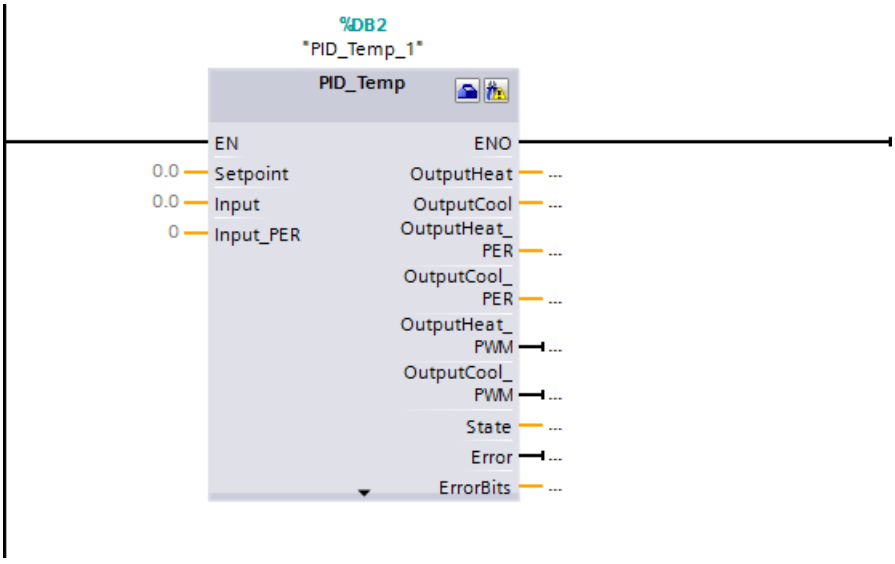
Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Количество ведомых устройств	"Config.Cascade.CountSlaves"	Int	255 >= Config.Cascade. CountSlaves >= 1	Количество ведомых регуляторов, получающих свое заданное значение непосредственного от главного регулятора. Инструкция PID_Temp обрабатывает это значение, наряду с другими, для функции анти-насыщения. "Количество ведомых устройств" доступно только в том случае, если флажок "Это регулятор является ведущим устройством" установлен (Config.Cascade.IsMaster = TRUE).
Это регулятор является ведомым устройством	"Config.Cascade.IsSlave"	Bool	Bool	Показывает, является ли этот регулятор ведомым устройством в каскаде. При установке флажка параметр "Config.Cascade.IsSlave" устанавливается на TRUE, вместо FALSE, если флажок не установлен (= по умолчанию). Необходимо установить флажок на странице свойств, чтобы разблокировать выпадающий список "SelectionMaster".

Пример: Каскадирование регуляторов

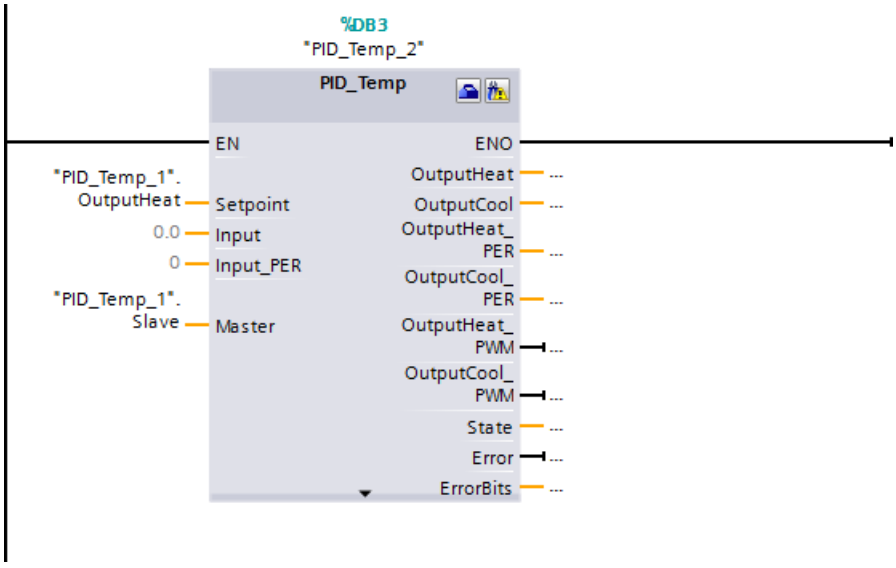
В следующем диалоговом окне "Основные настройки" представлен раздел "Входные / выходные параметры" и раздел "Каскад" для ведомого регулятора "PID_Temp_2" после выбора "PID_Temp_1" в качестве ведущего устройства. Устанавливаются соединения между ведущим и ведомым регуляторами:



Сегмент 1: В этих сегментах устанавливается соединение между ведущим устройством "PID_Temp_1" и ведомым устройством "PID_Temp_2" в редакторе текстов программы:



Сегмент 2: Устанавливается соединение между параметрами "OutputHeat" и "Slave" ведущего устройства "PID_Temp_1" и параметрами "Setpoint" и "Master" ведомого устройства "PID_Temp_2":



Автонастройка температурного процесса

Инструкция PID_Temp предлагает два режима работы для автоматической настройки:

- "Предварительная настройка" (параметр "Mode" = 1)
- "Точная регулировка" (параметр "Mode" = 2)

В зависимости от конфигурации регулятора доступны различные варианты этих методов настройки:

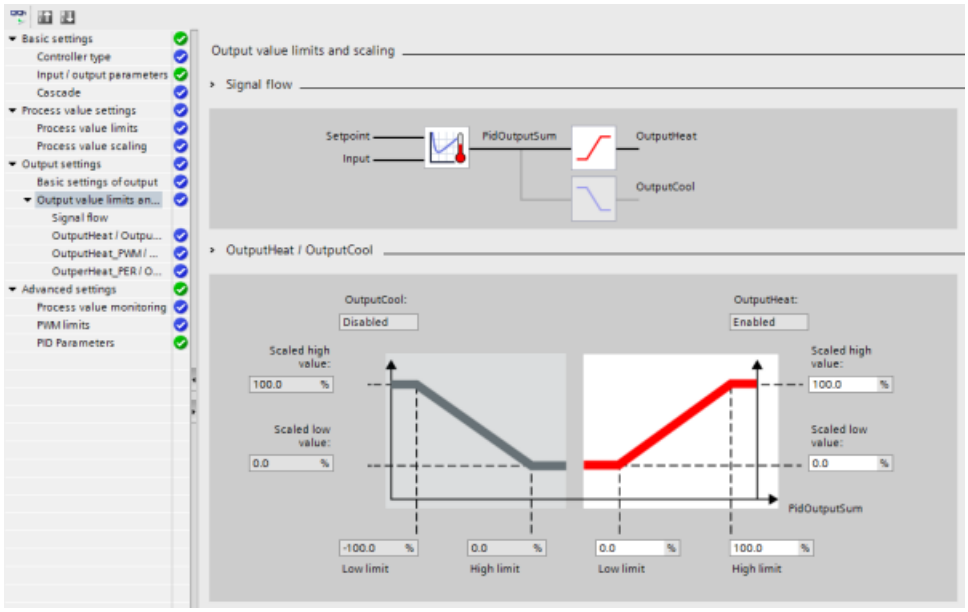
Конфигурация	**Регулятор с выходом нагрева**	**Регулятор с выходом нагрева и охлаждения и использование коэффициента охлаждения**	**Регулятор с выходом нагрева и охлаждения и использование два набора ПИД-параметров**
Соответствующие значения TO-DB	• Config.ActivateCooling = FALSE • Config.AdvancedCooling = не имеет значения	• Config.ActivateCooling = TRUE • Config.AdvancedCooling = FALSE	• Config.ActivateCooling = TRUE • Config.AdvancedCooling = TRUE
Доступные методы настройки	• "Первичная оптимизация нагрева" • "Точная регулировка нагрева" (нельзя использовать смещение охлаждения)	• "Первичная оптимизация нагрева" • "Точная регулировка нагрева" (можно использовать смещение охлаждения)	• "Первичная оптимизация нагрева и охлаждения" • "Первичная оптимизация нагрева" • "Первичная оптимизация охлаждения" • "Точная регулировка нагрева" (можно использовать смещение охлаждения) • "Точная регулировка охлаждения" (можно использовать смещение нагрева)

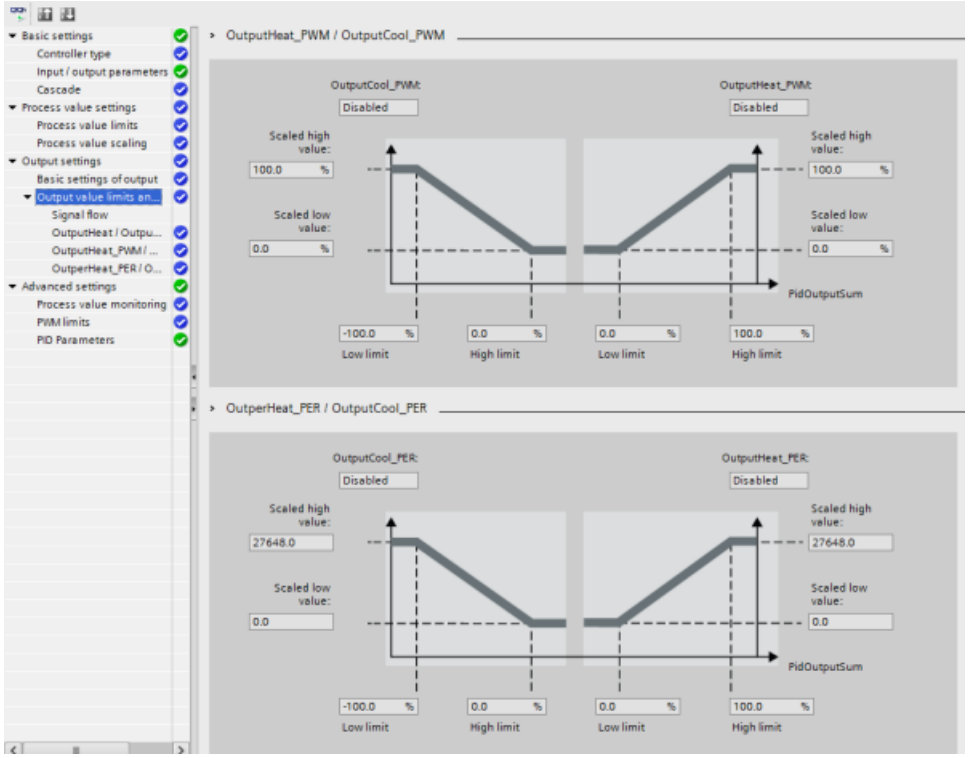
Пределы и масштабирование выходного значения

Активация охлаждения отключена

Если инструкция PID_Temp конфигурируется в качестве ведущего устройства для каскада, то флажок "Активировать выход (охлаждение)" в представлении "Основные настройки" снят и отключен, все настройки в представлении "Настройки выхода", которые зависят от активации охлаждения, также отключены.

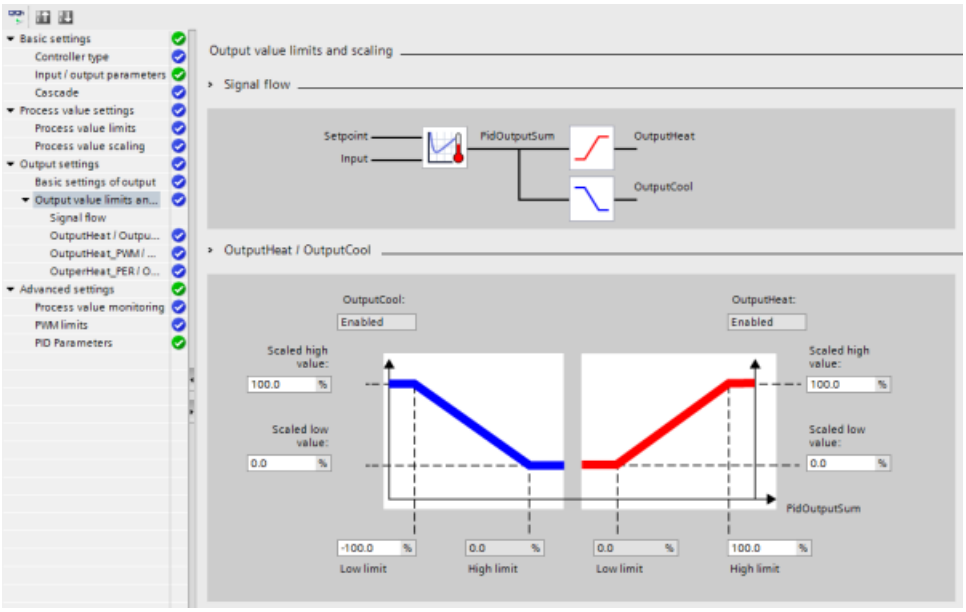
На рисунке ниже показан раздел "Пределы и масштабирование выходного значения" в представлении "Настройки выхода" с отключенным охлаждением (OutputHeat_PWM выбран в представлении "Входные / выходные параметры", а OutputHeat всегда включен):

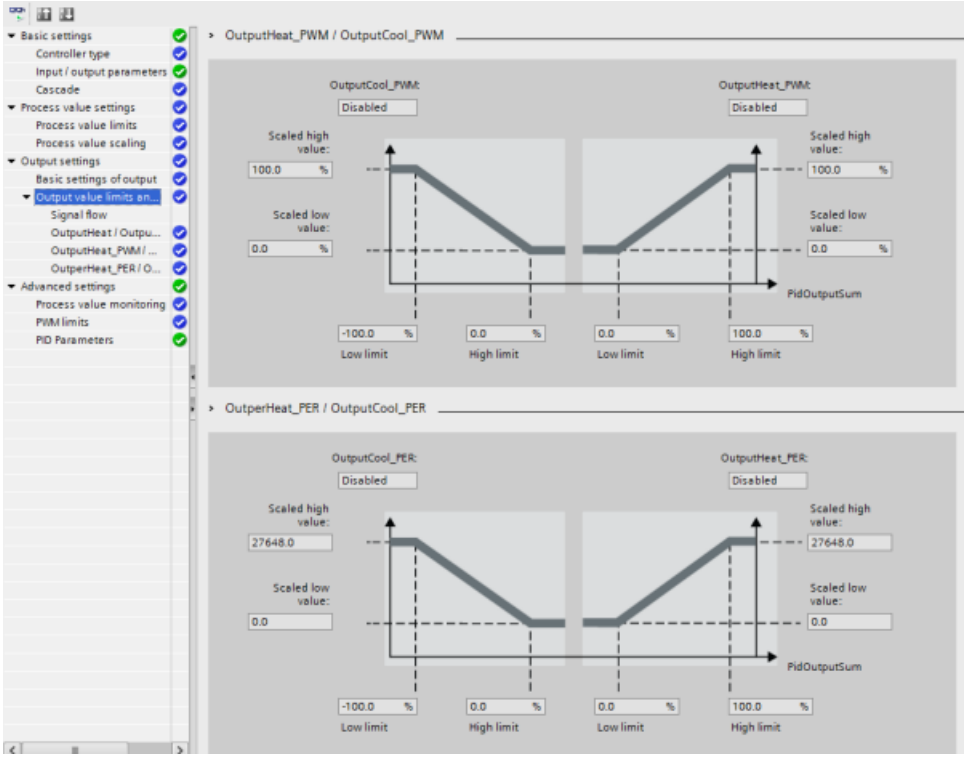




Активация охлаждения включена

На рисунке ниже показан раздел "Пределы и масштабирование выходного значения" в представлении "Настройки выхода" с включенным охлаждением (OutputCool_PER и OutputHeat_PWM выбраны в представлении "Входные / выходные параметры"; OutputCool и OutputHeat всегда включены):





Режимы работы

Для того, чтобы вручную изменить режим работы, пользователь должен установить параметр "Mode" регулятора и активировать его, изменив значение "ModeActivate" с FALSE на TRUE (активация по положительному фронту). Следует сбросить "ModeActivate" перед следующей сменой режимов работы; параметр не сбрасывается автоматически.

Выходной параметр "State" показывает текущий режим работы и переводится, если возможно, в требуемый "Mode". Параметр "State" не может быть изменен напрямую; он может быть изменен только через параметр "Mode" или через автоматическую смену режимов работы регулятора.

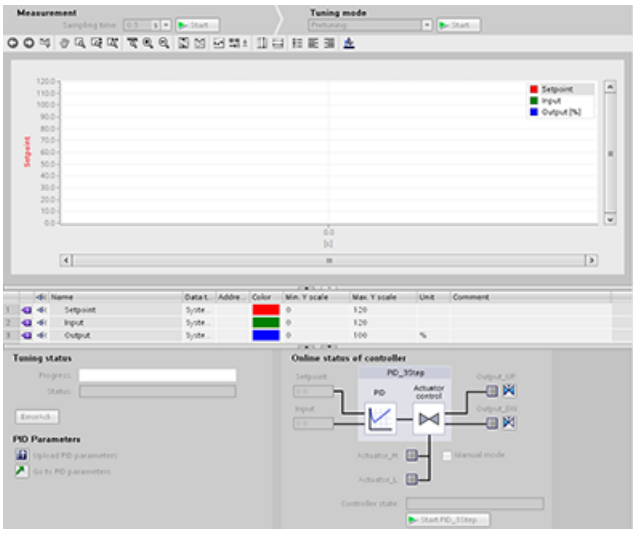
"Mode" / "State"	Название	Описание
0	Неактивно	Инструкция PID_Temp: • Деактивирует ПИД-алгоритм и широтно-импульсную модуляцию • Устанавливает все выходы регуляторов (OutputHeat, OutputCool, OutputHeat_PWM, OutputCool_PWM, OutputHeat_PER, OutputCool_PER), независимо от сконфигурированных выходных пределов или смещений, на "0" (FALSE). Переход в этот режим выполняется по установке "Mode" = 0, "Reset" = TRUE или через ошибку.
1	Предварительная настройка (настройка при запуске/ SUT)	Этот режим работы определяет параметры во время первого запуска регулятора. В отличие от PID_Compact, для PID_Temp с помощью параметров "Heat.EnableTuning" и "Cool.EnableTuning" необходимо выбрать, нужна ли настройка нагрева, настройка охлаждения или обе эти настройки. Можно активировать "предварительную настройку" из неактивного, автоматического или ручного режима. Если настройка прошла успешно, то PID_Temp переключается в автоматический режим. Если настройка завершилась неудачно, переключение рабочего режима зависит от “ActivateRecoverMode”.
2	Точная регулировка (настройка в Run/TIR)	Этот режим работы определяет оптимальное параметрирование ПИД-регулятора на заданном значении. В отличие от PID_Compact, для PID_Temp с помощью параметров "Heat.EnableTuning" и "Cool.EnableTuning" необходимо выбрать, нужна ли настройка нагрева или настройка охлаждения. Можно активировать "точную регулировку" из неактивного, автоматического или ручного режима. Если настройка прошла успешно, то PID_Temp переключается в автоматический режим. Если настройка завершилась неудачно, переключение рабочего режима зависит от “ActivateRecoverMode”.
3	Автоматический режим	В автоматическом режиме (стандартный режим ПИД-регулятора) результат ПИД-алгоритма определяет выходные значения. PID_Temp переключается в неактивный режим, если происходит ошибка и “ActivateRecoverMode” = FALSE. Если ошибка происходит и “ActivateRecoverMode” = TRUE, переключение рабочего режима зависит от ошибки. Обратиться к параметрам ErrorBit (Страница 624) инструкции PID_Temp для получения дополнительной информации.

"Mode" / "State"	Название	Описание
4	Ручной режим	В этом режиме ПИД-регулятор масштабирует, ограничивает и передает значение параметра “ManualValue” на выходы. ПИД-регулятор назначает “ManualValue” ы масштабировании ПИД-алгоритма (как “PidOutputSum”), так что его значение определяет, будет ли активирован выход нагрева или охлаждения. Переход в этот режим выполняется по установке "Mode" = 4 или "ManualEnable" = TRUE.
5	Замещающее выходное значение с контролем ошибок (режим восстановления)	Переход в этот режим выполняется по установке "Mode" = 5. Режим представляет собой автоматическую реакцию регулятора на ошибку, если автоматический режим активен в момент возникновения ошибки: • SetSubstituteOutput = FALSE (последнее действительное выходное значение) • SetSubstituteOutput = TRUE (сохраненное в параметре "SubstituteOutput" значение) Когда PID_Temp находится в “Автоматическом режиме” и параметр “ActivateRecoverMode” = TRUE, PID_Temp переходит в этот режим в случае следующих ошибок: • "Недействительное значение в параметре "Input_PER". Проверить наличие ошибки на аналоговом входе (например, обрыв провода)". (ErrorBits = DW#16#0002) • "Недействительное значение в параметре "Input". Значение не является числом." (ErrorBits = DW#16#0200) • "Расчет выходного значения не удался. Проверить параметры ПИД." (ErrorBits = DW#16#0400) • "Недействительное значение в параметре "Setpoint". Значение не является числом." (ErrorBits = DW#16#1000) Когда ошибка будет устранена, PID_Temp снова переходит в автоматический режим.

10.2.7 Ввод в эксплуатацию регуляторов PID_Compact и PID_3Step

Редактор ввода в эксплуатацию используется для конфигурирования ПИД-регулятора на автоматическую настройку при запуске и на автоматическую настройку во время работы. Чтобы открыть редактор ввода в эксплуатацию, кликнуть по соответствующему значку в папке инструкций или в дереве проекта.

Таблица 10-33 Пример экрана ввода в эксплуатацию (PID_3Step)

	• Измерение: Для того, чтобы показать заданное значение, технологическое значение (входное значение) и выходное значение на тренде реального времени, следует ввести время выборки и нажать кнопку "Start". • Режим настройки: Для настройки ПИД-регулятора следует выбрать "Предварительную настройку" или "Точную регулировку" (ручную) и нажать кнопку "Start". ПИД-регулятор выполняет несколько операций, чтобы рассчитать время реакции системы и время обновления. Соответствующие параметры настройки рассчитываются из этих значений. После завершения процесса настройки можно сохранить новые параметры, нажав кнопку "Выгрузить параметры ПИД" в разделе "Параметры ПИД" редактора ввода в эксплуатацию Если во время настройки происходит ошибка, выходное значение ПИД переходит в 0. ПИД-регулятор переводится в "неактивный" режим работы. Состояние показывает ошибку.

Управление стартовыми значениями ПИД

Можно отредактировать фактические значения параметров конфигурации ПИД так, чтобы поведение ПИД-регулятора могло быть оптимизировано в режиме онлайн.

Открыть "Технологические объекты" для своего ПИД-регулятора и его объект "Конфигурация". Чтобы получить доступ к управлению стартовыми значениями, кликнуть по "значку очков" в левом верхнем углу диалогового окна:



Теперь можно изменить значение любого из параметров конфигурации ПИД-регулятора, как показано на рисунке ниже.

Можно сравнить фактическое значение со стартовым значение в проекте (офлайн) и стартовым значением PLC (онлайн) для каждого параметра. Это необходимо, чтобы выявить онлайн/офлайн различия блока данных технологического объекта (TO-DB) и знать значения, которые будут использоваться в качестве текущих значений при следующем переходе PLC из STOP в START. Кроме того, значок сравнения предоставляет визуальную индикацию для упрощения обнаружения различий в онлайн/офлайне:

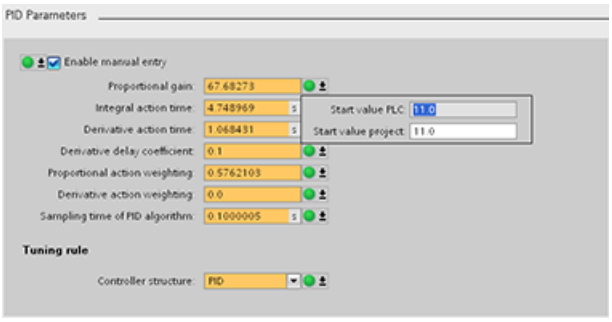
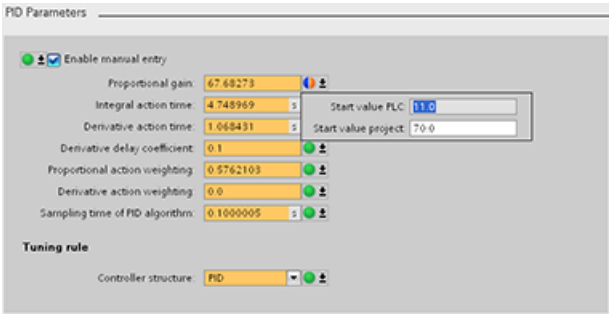


Рисунок выше представляет экран параметров ПИД со значками сравнения. Символы показывают, какие из значений различаются в онлайн и офлайн проектах. Зеленый значок обозначает одинаковые значения. Синий/оранжевый значок указывает на то, что значения отличаются друг от друга.

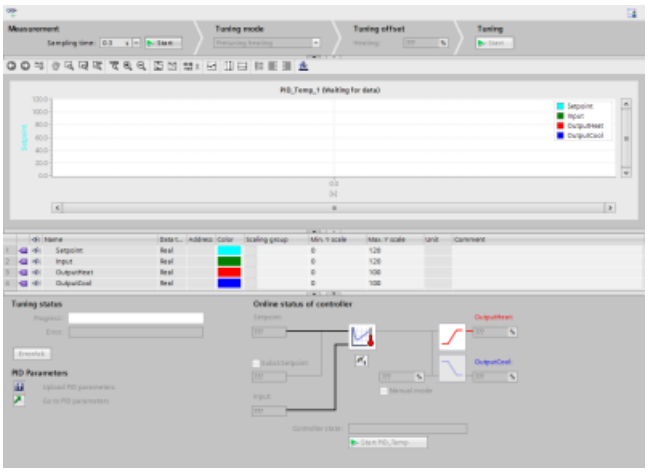
Дополнительно можно нажать кнопку параметра с направленной вниз стрелкой, чтобы открыть небольшое окно со стартовым значением проекта (офлайн) и стартовым значением PLC (онлайн) для каждого параметра:



10.2.8 Ввод в эксплуатацию регулятора PID_Temp

Редактор ввода в эксплуатацию используется для конфигурирования ПИД-регулятора на автоматическую настройку при запуске и на автоматическую настройку во время работы. Чтобы открыть редактор ввода в эксплуатацию, кликнуть по соответствующему значку в паке инструкций или в дереве проекта.

Таблица 10-34 Пример экрана ввода в эксплуатацию (PID_Temp)

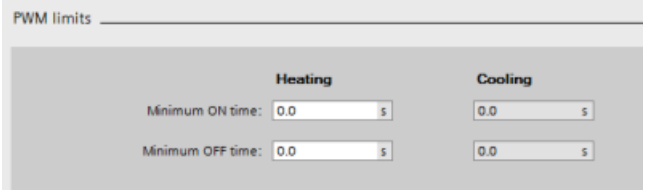
	Измерение: Для того, чтобы показать заданное значение, технологическое значение (входное значение) и выходное значение на тренде реального времени, следует ввести время выборки и нажать кнопку "Start". Режим настройки: Для настройки PID_Temp регулятора следует выбрать "Предварительную настройку" или "Точную регулировку" (ручную) и нажать кнопку "Start". ПИД-регулятор выполняет несколько операций, чтобы рассчитать время реакции системы и время обновления. Соответствующие параметры настройки рассчитываются из этих значений. После завершения процесса настройки можно сохранить новые параметры, нажав кнопку "Выгрузить параметры ПИД" в разделе "Параметры ПИД" редактора ввода в эксплуатацию Если во время настройки происходит ошибка, выходное значение ПИД переходит в 0. ПИД-регулятор переводится в "неактивный" режим работы. Состояние показывает ошибку.

Предельные значения PWM

Для исполнительных элементов, регулируемых с помощью программной PWM функции регулятора PID_Temp, может потребоваться защита от слишком короткой длительности импульса (например, тиристорный переключатель должен быть включен больше чем на 20 мс, прежде чем он сможет реагировать). Для этого устанавливается минимальное время включённого состояния. Исполнительный элемент может игнорировать короткие импульсы, ухудшая тем самым качество регулирования. Здесь может оказаться необходимым минимальное время выключенного состояния (например, чтобы предотвратить перегрев).

Для просмотра "Ограничений PWM", следует открыть функциональное представление в конфигурации технологических объектов (ТО) и выбрать в дереве переходов в "Расширенных настройках" пункт "Ограничения PWM".

После открытия представления “Ограничения PWM" в функциональном представлении и активации контроля (значок “очки”), все элементы управления отображают онлайн значение контроля из TO-DB с оранжевым фоном и многозначное регулирование, и теперь можно редактировать значения (если условия конфигурации выполнены, см. таблицу ниже).



Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Минимальное время включённого состояния (нагрев) [1,2]	"Config.Output.Heat.MinimumOnTime"	Real	100000,0 >= "Config.Output.Heat.MinimumOnTime >= 0,0	Импульс на OutputHeat_PWM всегда длиннее, чем это значение.
Минимальное время выключённого состояния (нагрев) [1,2]	"Config.Output.Heat.MinimumOffTime"	Real	100000,0 >= "Config.Output.Heat.MinimumOffTime >= 0,0	Прерывание на OutputHeat_PWM всегда длиннее, чем это значение.
Минимальное время включённого состояния (охлаждение) [1,3,4]	"Config.Output.Cool.MinimumOnTime"	Real	100000,0 >= Config.Output.Cool.MinimumOnTime >= 0,0	Импульс на OutputCool_PWM всегда длиннее, чем это значение.
Минимальное время выключённого состояния (охлаждение) [1,3,4]	"Config.Output.Cool.MinimumOffTime"	Real	100000,0 >= Config.Output.Cool.MinimumOffTime >= 0,0	Прерывание на OutputCool_PWM всегда длиннее, чем это значение.

[1] Поле отображает значение в единицах времени "с" (секунды).

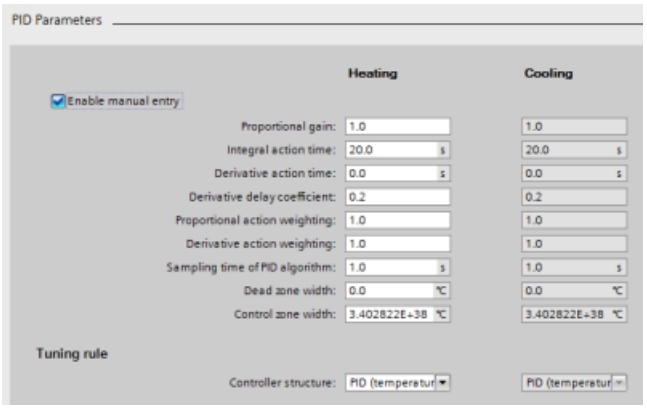
[2] Если для "Выбора выхода (нагрев)" в представлении "Основные настройки" установлено не "OutputHeat_PWM" (Config.Output.Heat.Select = TRUE), то это значение должно быть установлено на "0,0".

[3] Если для "Выбора выхода (охлаждение)" в представлении "Основные настройки" установлено не "OutputCool_PWM" (Config.OutputCool.Select = TRUE), то это значение должно быть установлено на "0,0".

[4] Доступно, только если в представлении "Основные настройки" включить "Активировать выход (охлаждение)" (Config.ActivateCooling = TRUE).

Параметры ПИД

Представление “Расширенные настройки”, раздел "Параметры ПИД” показано ниже с деактивированной функцией охлаждения и/или “Переключение параметров ПИД”.



Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Активировать ручной вывод	"Retain.CtrlParams. SetByUser"	Bool	Bool	Следует установить этот флажок, чтобы можно было вводить параметры ПИД вручную
Пропорциональное усиление (нагрев) [2]	"Retain.CtrlParams. Heat.Gain"	Real	Усиление >= 0,0	Пропорциональное усиление ПИД для нагрева.
Время интегрирования (нагрев) [1,2]	"Retain.CtrlParams. Heat.Ti"	Real	100000,0 >= Ti >= 0,0	Время интегрирования ПИД для нагрева.
Время дифференцирования (нагрев) [1,2]	"Retain.CtrlParams. Heat.Td"	Real	100000,0 >= Td >= 0,0	Время дифференцирования ПИД для нагрева.
Коэффициент задержки дифференцирования (нагрев) [2]	"Retain.CtrlParams. Heat.TdFiltRatio"	Real	TdFiltRatio >= 0,0	Коэффициент задержки дифференцирования для нагрева, определяющий время запаздывания дифференцирования как коэффициенты из времени дифференцирования ПИД.

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Оценка П-составляющей (нагрев) [2]	"Retain.CtrlParams.Heat.PWeighting"	Real	1,0 >=PWeighting >= 0,0	Оценка пропорционального усиления ПИД для нагрева в прямом или обратном контуре регулирования.
Оценка Д-составляющей (нагрев) [2]	"Retain.CtrlParams.Heat.DWeighting"	Real	1,0 >=DWeighting >= 0,0	Оценка дифференцирующей составляющей ПИД для нагрева в прямом или обратном контуре регулирования.
Время выборки ПИД-алгоритма (нагрев) [1,2]	"Retain.CtrlParams.Heat.Cycle"	Real	100000,0 >=Цикл > 0,0	Внутренний цикл вызова ПИД-регулятора для нагрева. С округлением до целого кратного времени цикла вызова FB.
Ширина мертвой зоны (нагрев) [2,3]	"Retain.CtrlParams.Heat.DeadZone"	Real	DeadZone >= 0,0	Ширина мертвой зоны для погрешности регулирования нагрева.
Зона регулирования (нагрев) [2,3]	"Retain.CtrlParams.Heat.ControlZone"	Real	ControlZone > 0,0	Ширина зоны погрешности регулирования для нагрева, где активен ПИД-регулятор. При выходе погрешности регулирования из этой зоны, выход переключается на максимальные выходные значения. Значение по умолчанию "MaxReal", т.е. зона регулирования отключена до тех пор, пока не выполняется автоматическая настройка. Значение "0,0" для зоны регулирования недопустимо. При значении "0,0" регулятор PID_Temp ведет себя как двухпозиционный регулятор, который всегда выполняет нагрев или охлаждение на полной мощности.

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Структура регулятора (нагрев)	"PIDSelfTune.SUT. TuneRuleHeat", "PIDSelfTune.TIR. TuneRuleHeat"	Int	"PIDSelfTune.SUT. TuneRuleHeat" = 0..2, "PIDSelfTune.TIR. TuneRuleHeat" = 0..5	Здесь можно выбрать алгоритм настройки для нагрева. Возможный выбор: • ПИД (температура) (= по умолчанию) ("PIDSelfTune.SUT. TuneRuleHeat" = 2) ("PIDSelfTune.TIR. TuneRuleHeat" = 0) • ПИД ("PIDSelfTune.SUT. TuneRuleHeat" = 0) ("PIDSelfTune.TIR. TuneRuleHeat" = 0) • ПИ ("PIDSelfTune.SUT. TuneRuleHeat" = 1) ("PIDSelfTune.TIR. TuneRuleHeat" = 4) Любая другая комбинация показывает “Определено пользователем”, но “Определено пользователем” по умолчанию отсутствует. Выбор “ПИД (температура)” является новым для PID_Temp со специфическим методом предварительной настройки (SUT) для температурных процессов.
Пропорциональное усиление (охлаждение) [4]	"Retain.CtrlParams. Cool.Gain"	Real	Усиление >= 0,0	Пропорциональное усиление ПИД для охлаждения.
Время интегрирования (охлаждение) [1,4]	"Retain.CtrlParams. Cool.Ti"	Real	100000,0 >=Ti >= 0,0	Время интегрирования ПИД для охлаждения.
Время дифференцирования (охлаждение) [1,4]	"Retain.CtrlParams. Cool.Td"	Real	100000,0 >=Td >= 0,0	Время дифференцирования ПИД для охлаждения.

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Коэффициент задержки дифференцирования (охлаждение) [4]	Retain.CtrlParams. Cool.TdFiltRatio"	Real	TdFiltRatio >= 0,0	Коэффициент задержки дифференцирования для охлаждения, определяющий задержку дифференцирования как коэффициенты из времени дифференцирования ПИД.
Оценка П-составляющей (охлаждение) [4]	"Retain.CtrlParams. Cool.PWeighting"	Real	1,0 >=PWeighting >= 0,0	Оценка пропорционального усиления ПИД для охлаждения в прямом или обратном контуре регулирования.
Оценка Д-составляющей (охлаждение) [4]	Retain.CtrlParams. Cool.DWeighting"	Real	1,0 >=DWeighting >= 0,0	Оценка дифференцирующей составляющей ПИД для охлаждения в прямом или обратном контуре регулирования.
Время выборки ПИД-алгоритма (охлаждение) [1,4]	"Retain.CtrlParams. Cool.Cycle"	Real	100000,0 >=Цикл > 0,0	Внутренний цикл вызова ПИД-регулятора для охлаждения. С округлением до целого кратного времени цикла вызова FB.
Ширина мертвой зоны (охлаждение) [3,4]	"Retain.CtrlParams. Cool.DeadZone"	Real	DeadZone >= 0,0	Ширина мертвой зоны для погрешности регулирования охлаждения.

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Зона регулирования (охлаждение) [3,4]	"Retain.CtrlParams.Cool.ControlZone"	Real	ControlZone > 0,0	Ширина зоны погрешности регулирования для охлаждения, где активен ПИД-регулятор. При выходе погрешности регулирования из этой зоны, выход переключается на максимальные выходные значения. Значение по умолчанию "MaxReal", т.е. зона регулирования отключена до тех пор, пока не выполняется автоматическая настройка. Значение "0,0" для зоны регулирования недопустимо. При значении "0,0" регулятор PID_Temp ведет себя как двухпозиционный регулятор, который всегда выполняет нагрев или охлаждение на полной мощности.

Настройка	TO-DB параметр	Тип данных	Диапазон значений	Описание
Структура регулятора (охлаждение)	"PIDSelfTune.SUT. TuneRuleCool", "PIDSelfTune.TIR. TuneRuleCool"	Int	"PIDSelfTune.SUT. TuneRuleHeat" = 0..2, "PIDSelfTune.TIR. TuneRuleHeat" = 0..5	Здесь можно выбрать алгоритм настройки для охлаждения. Возможный выбор: • ПИД (температура) (= по умолчанию) ("PIDSelfTune.SUT. TuneRuleCool" = 2) ("PIDSelfTune.TIR. TuneRuleCool = 0) • ПИД ("PIDSelfTune.SUT. TuneRuleCool" = 0) ("PIDSelfTune.TIR. TuneRuleCool" = 0) • ПИ ("PIDSelfTune.SUT. TuneRuleCool" = 1) ("PIDSelfTune.TIR. TuneRuleCool" = 4) Любая другая комбинация показывает “Определено пользователем”, но “Определено пользователем” по умолчанию отсутствует. Выбор “ПИД (температура)” является новым для PID_Temp со специфическим методом предварительной настройки (SUT) для температурных процессов. Доступно только при выборе следующих элементов: "Активировать выход (охлаждение)" в представлении "Основные настройки" ("Config.ActivateCooling = TRUE) и "Переключение параметров ПИД" в представлении "Настройки выхода" (Config.AdvancedCooling = TRUE).

[1] Поле отображает значение в единицах времени "c" (секунды).

[2] Доступно, только если в ПИД параметрах включить "Активировать ручной вывод" ("Retain.CtrlParams.SetByUser" = TRUE).

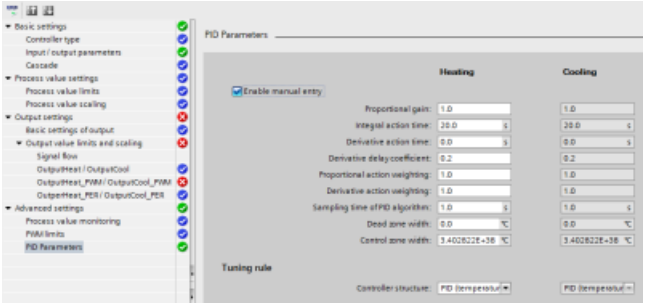
[3] Единица измерения, выбранная в представлении “Основные настройки”, отображается в конце поля.

[4] Доступно только при выборе следующих элементов: "Активировать ручной ввод" в ПИД параметрах ("Retain.CtrlParams.SetByUser" = TRUE), "Активировать выход (охлаждение)" в представлении "Основные настройки" ("Config.ActivateCooling = TRUE) и "Переключение параметров ПИД" в представлении "Настройки выхода" (Config.AdvancedCooling = TRUE).

Управление стартовыми значениями ПИД

Можно отредактировать фактические значения параметров конфигурации ПИД так, чтобы поведение ПИД-регулятора могло быть оптимизировано в режиме онлайн.

Открыть "Технологические объекты" для своего ПИД-регулятора и его объект "Конфигурация". Чтобы получить доступ к управлению стартовыми значениями, кликнуть по "значку очков" в левом верхнем углу диалогового окна:



Теперь можно изменить значение любого из параметров конфигурации ПИД-регулятора, как показано на рисунке ниже.

Можно сравнить фактическое значение со стартовым значение в проекте (офлайн) и стартовым значением PLC (онлайн) для каждого параметра. Это необходимо, чтобы выявить онлайн/офлайн различия блока данных технологического объекта (TO-DB) и знать значения, которые будут использоваться в качестве текущих значений при следующем переходе PLC из STOP в START. Кроме того, значок сравнения предоставляет визуальную индикацию для упрощения обнаружения различий в онлайн/офлайне:

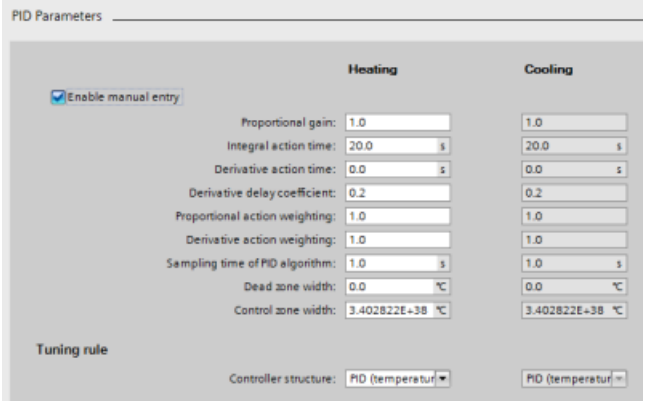
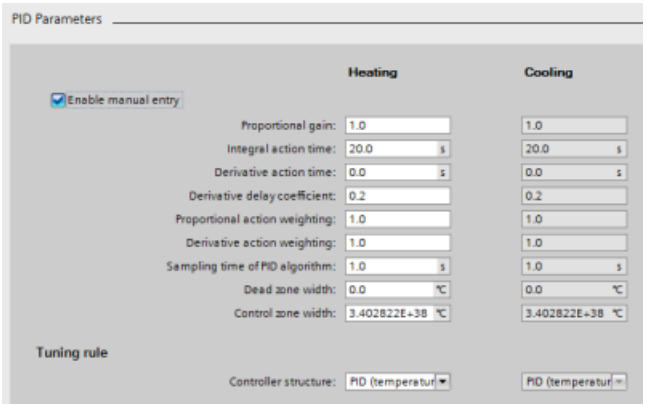


Рисунок выше представляет экран параметров ПИД со значками сравнения. Символы показывают, какие из значений различаются в онлайн и офлайн проектах. Зеленый значок обозначает одинаковые значения. Синий/оранжевый значок указывает на то, что значения отличаются друг от друга.

Дополнительно можно нажать кнопку параметра с направленной вниз стрелкой, чтобы открыть небольшое окно со стартовым значением проекта (офлайн) и стартовым значением PLC (онлайн) для каждого параметра:



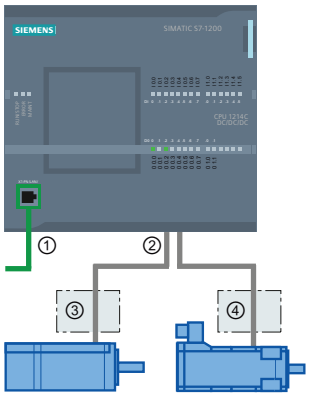
10.3 Управление перемещением:

CPU предлагает функции управления перемещением для работы с шаговыми двигателями и серводвигателями с импульсным интерфейсом. Функция управления перемещением выполняет управление и мониторинг приводов.

- Технологический объект "Ось" конфигурирует данные механического привода, интерфейс привода, динамические параметры и другие свойства привода.
- Пользователь конфигурирует выходы CPU для импульсов и направления для управления приводом.
- Программа пользователя использует инструкции управления перемещением для управления осью и запуска задач движения.
- Через интерфейс PROFINET можно установить интерактивное соединение между CPU и программатором. Помимо интерактивных функций CPU, для управления перемещением доступны и другие функции ввода в эксплуатацию и диагностики.

Примечание

Изменения, вносимые в конфигурацию управления перемещением и загружаемые в рабочем состоянии RUN в целевую систему, не вступают в силу, пока CPU не перейдет из режима STOP в режим RUN.



① PROFINET
② Выходы импульсов и направления
③ Силовая часть для шагового двигателя
④ Силовая часть для серводвигателя

У DC/DC/DC вариантов S7-1200 CPU есть встроенные выходы для прямого управления приводами. Релейные варианты CPU требуют сигнальной платы с DC выходами для управления приводами.

Сигнальная плата (SB) расширяет встроенный ввод-вывод путем добавления нескольких дополнительных входов и выходов. SB с двумя цифровыми выходами может использоваться в качестве выходов импульсов и направления для управления двигателем. SB с четырьмя цифровыми выходами может использоваться в качестве выходов импульсов и направления для управления двумя двигателями. Встроенные релейные выходы не могут использоваться в качестве импульсных выходов для управления двигателями. Независимо от того, используются ли встроенные I/O, I/O на SB или комбинация обоих этих вариантов, может быть определено максимум четыре импульсных генератора.

Четырех импульсных генератора предлагают назначения ввода-вывода по умолчанию. Однако генераторы импульсов могут быть сконфигурированы и для любого цифрового выхода на CPU или на SB. Импульсные генераторы на CPU не могут быть назначены SM или распределенному вводу-выводу.

Примечание

Шаговый выходной интерфейс не может использоваться другими инструкциями в программе пользователя

Если выходы CPU или сигнальной платы конфигурируются как генераторы импульсов (для PWM или инструкций управления перемещением), то соответствующие адреса вывода больше не управляют выходами. Если программа пользователя записывает значение на выход, который используется как генератор импульсов, то CPU не записывает это значение на физический выход.

Таблица 10-35 Максимальное количество управляемых приводов

Тип CPU		Встроенные I/O; SB не установлена		С одной SB (2 x DC-выходы)		С одной SB (4 x DC-выходы)	
		С направлением	Без направления	С направлением	Без направления	С направлением	Без направления
CPU 1211C	DC/DC/DC	2	4	3	4	4	4
	AC/DC/реле	0	0	1	2	2	4
	DC/DC/реле	0	0	1	2	2	4

Тип CPU		Встроенные I/O; SB не установлена		С одной SB (2 x DC-выходы)		С одной SB (4 x DC-выходы)	
		С направлением	Без направления	С направлением	Без направления	С направлением	Без направления
CPU 1212C	DC/DC/DC	3	4	3	4	4	4
	AC/DC/реле	0	0	1	2	2	4
	DC/DC/реле	0	0	1	2	2	4
CPU 1214C	DC/DC/DC	4	4	4	4	4	4
	AC/DC/реле	0	0	1	2	2	4
	DC/DC/реле	0	0	1	2	2	4
CPU 1215C	DC/DC/DC	4	4	4	4	4	4
	AC/DC/реле	0	0	1	2	2	4
	DC/DC/реле	0	0	1	2	2	4
CPU 1217C	DC/DC/DC	4	4	4	4	4	4

Примечание

Возможно до четырех генераторов импульсов.

Независимо от того, используются ли встроенные I/O, I/O на SB или комбинация обоих этих вариантов, может быть определено максимум четыре импульсных генератора.

Таблица 10-36 Выход CPU: Максимальная частота

CPU	Выходной канал CPU:	Выход импульсов и направления	A/B, вперед/назад и импульс/направление
1211C	Qa.0 до Qa.3	100 кГц	100 кГц
1212C	Qa.0 до Qa.3	100 кГц	100 кГц
	Qa.4, Qa.5	20 кГц	20 кГц
1214C и 1215C	Qa.0 до Qa.3	100 кГц	100 кГц
	Qa.4 до Qb.1	20 кГц	20 кГц
1217C	DQa.0 до DQa.3 (.0+, .0- до .3+, .3-)	1 МГц	1 МГц
	DQa.4 до DQa.1	100 кГц	100 кГц

Таблица 10-37 Выход сигнальной платы: Максимальная частота (опциональная плата)

Сигнальная плата (SB)	Выходной канал SB	Выход импульсов и направления	A/B, вперед/назад и импульс/направление
SB 1222, 200 кГц	DQe.0 до DQe.3	200 кГц	200 кГц
SB 1223, 200 кГц	DQe.0, DQe.1	200 кГц	200 кГц
SB 1223	DQe.0, DQe.1	20 кГц	20 кГц

Таблица 10-38 Пороговые частоты для выходов импульсов

Импульсный выход	**Частота**
Встроенный	4 PTO: 2 Гц ≤ f ≤ 1 МГц, 4 PTO: 2 Гц ≤ f ≤ 100 кГц или любая комбинация этих значений для 4 PTO.[1] [2]
Стандартная SB	2 Гц ≤ f ≤ 20 кГц
Высокоскоростные SB	2 Гц ≤ f ≤ 200 кГц

[1] См. таблицу ниже для четырех возможных комбинаций выходной скорости для CPU 1217C.

[2] См. таблицу ниже для четырех возможных комбинаций выходной скорости для CPU 1211C, CPU 1212C, CPU 1214C или CPU 1215C.

Пример: Конфигурации скорости импульсного выхода CPU 1217C

Примечание

Со встроенными дифференциальными выходами CPU 1217C может генерировать четыре импульсных выхода до 1 Мгц.

Следующие примеры показывают четыре возможные комбинации выходных скоростей:

- Пример 1: 4 x 1 МГц PTO, отсутствует выход направления
- Пример 2: 1 x 1 МГц, 2 x 100 кГц и 1 x 20 кГц PTO, все с выходами направления
- Пример 3: 4 x 200 кГц PTO, отсутствует выход направления
- Пример 4: 2 x 100 кГц PTO и 2 x 200 кГц PTO, все с выходами направления

P = импульс B = направление		**Встроенные выходы CPU**										**Выходы высокоскоростной SB**				**Выходы стандартной SB**	
		Выходы 1 МГц (Q)				Выходы 100 кГц (Q)						Выходы 200 кГц (Q)				Выходы 20 кГц (Q)	
		0,0+	0,1+	0,2+	0,3+	0,4	0,5	0,6	0,7	1,0	1,1	4,0	4,1	4,2	4,3	4,0	4,1
		0,0-	0,1-	0,2-	0,3-												
Пример 1: 4 x 1 МГц (отсутствует выход направления)	PTO1	P															
	PTO2		P														
	PTO3			P													
	PTO4				P												

P = импульс B = направление		**Встроенные выходы CPU**										**Выходы высокоскоростной SB**				**Выходы стандартной SB**	
Пример 2: 1 x 1 МГц, 2 x 100 кГц и 1 x 20 кГц (все с выходами направления)	PTO1	P	D														
	PTO2					P	D										
	PTO3							P	D								
	PTO4															P	D
Пример 3: 4 - 200 кГц (отсутствует выход направления)	PTO1											P					
	PTO2												P				
	PTO3													P			
	PTO4														P		
Пример 4: 2 - 100 кГц; 2 - 200 кГц (все с выходами направления)	PTO1					P	D										
	PTO2							P	D								
	PTO3											P	D				
	PTO4													P	D		

Пример: Конфигурации скорости импульсного выхода CPU 1211C, CPU 1212C, CPU 1214C и CPU 1215C

Следующие примеры показывают четыре возможные комбинации выходных скоростей:

- Пример 1: 4 x 100 кГц PTO, отсутствует выход направления
- Пример 2: 2 x 100 кГц PTO и 2 x 20 кГц PTO, все с выходами направления
- Пример 3: 4 x 200 кГц PTO, отсутствует выход направления
- Пример 4: 2 x 100 кГц PTO и 2 x 200 кГц PTO, все с выходами направления

P = импульс **B = направление**		**Встроенные выходы CPU**										**Выходы высокоскоростной SB**				**Выходы стандартной SB**	
		Выходы 100 кГц (Q)				Выходы 20 кГц (Q)						Выходы 200 кГц (Q)				Выходы 20 кГц (Q)	
		0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	1,0	1,1	4,0	4,1	4,2	4,3	4,0	4,1
		CPU 1211C															
		CPU 1212C				CPU 1212C											
		CPU 1214C				CPU 1214C		CPU 1214C									
		CPU 1215C				CPU 1215C		CPU 1215C									
Пример 1: 4 - 100 кГц (отсутствует выход направления)	PTO1	P															
	PTO2		P														
	PTO3			P													
	PTO4				P												
Пример 2: 2 x 100 кГц, 2 x 20 кГц (все с выходами направления)	PTO1	P	D														
	PTO2			P	D												
	PTO3					P	D										
	PTO4							P	D								
Пример 3: 4 - 200 кГц (отсутствует выход направления)	PTO1											P					
	PTO2												P				
	PTO3													P			
	PTO4														P		
Пример 4: 2 - 100 кГц; 2 - 200 кГц (все с выходами направления)	PTO1	P	D														
	PTO2			P	D												
	PTO3											P	D				
	PTO4													P	D		

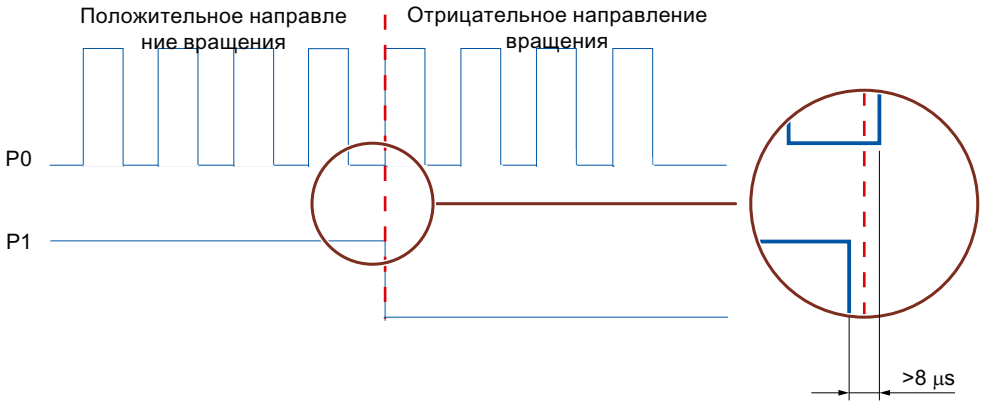
Примечание

Дополнительную информацию см. в *SIMATIC STEP 7 S7-1200 Motion Control V6.0 в TIA Portal V15*.

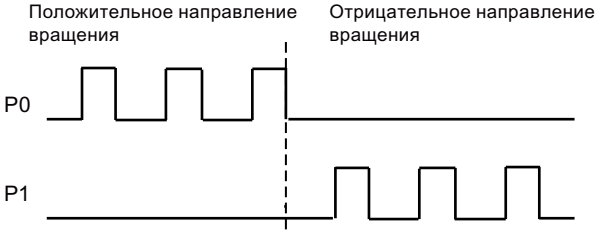
10.3.1 Фазирование

Интерфейс фазирования для шагового/сервопривода предлагает следующие четыре опции:

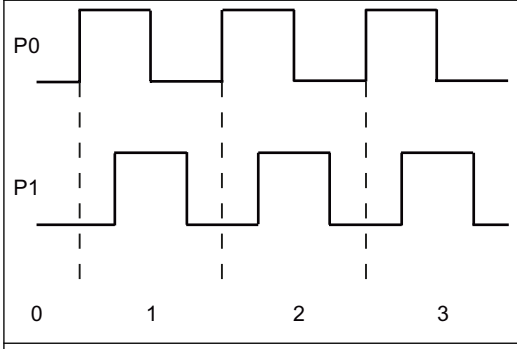
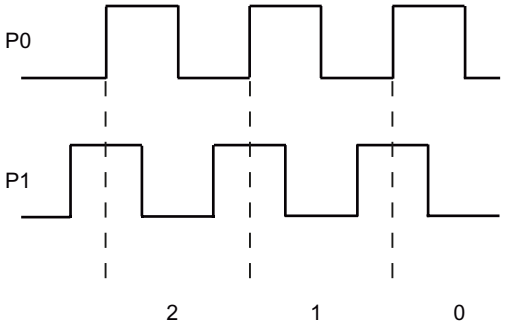
- PTO (импульс A и направление B): Если выбирается PTO опция (импульс A и направление B), то выход (P0) управляет импульсами, выход (P1) управляет направлением. P1 имеет высокий уровень (high, активный), если импульсы создаются в положительном направлении. P1 имеет низкий уровень (low , неактивный), если импульсы создаются в отрицательном направлении:



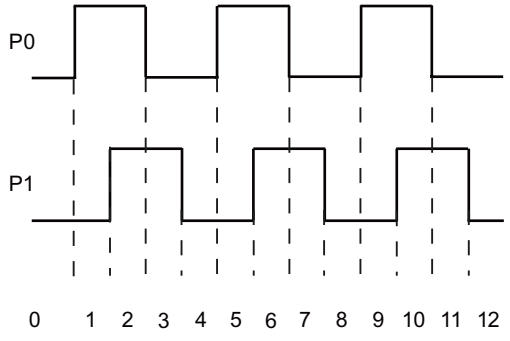
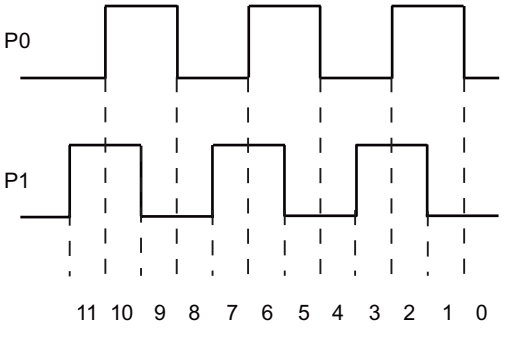
- PTO (импульс прямого счета A и импульс обратного счета B): Если выбирается PTO опция (счет в прямом направлении A и счет в обратном направлении B), то выход (P0) выводит импульсы для положительного направления, а другой выход (P1) выводит импульсы для отрицательного направления.



- PTO (A/B с фазовым сдвигом) Если выбирается PTO опция (A/B с фазовым сдвигом), то оба выхода выдают импульсы с указанной скоростью, но с фазовым сдвигом на 90 градусов. Это 1X конфигурация, означающая, что один импульс соответствует промежутку времени между положительными переходами P0. В этом случае направление определяется по тому выходу, фронт которого первым изменится с 0 на 1. P0 опережает P1 для положительного направления. P1 опережает P0 для отрицательного направления.
  Количество сгенерированных импульсов базируется на числе переходов от 0 к 1 фазы A. Фазовое соотношение определяет направление перемещения:

PTO (A/B с фазовым сдвигом)	
Фаза A опережает фазу B (перемещение в положительном направлении)	Фаза A следует за фазой B (перемещение в отрицательном направлении)
	
Количество импульсов	Количество импульсов

- PTO (A/B с четырехкратным фазовым сдвигом): Если выбирается PTO опция (A/B с четырехкратным фазовым сдвигом), то оба выхода выдают импульсы с указанной скоростью, но с фазовым сдвигом на 90 градусов. Четырехкратная опция - это 4X конфигурация, в которой один импульс соответствует переходу всех выходов (положительному и отрицательному). В этом случае направление определяется по тому выходу, фронт которого первым изменится с 0 на 1. P0 опережает P1 для положительного направления. P1 опережает P0 для отрицательного направления. Четырехкратная опция базируется на положительных и отрицательных переходах обеих фаз A и B. Число переходов конфигурируется пользователем. Фазовое соотношение (A опережает B или B опережает A) определяет направление перемещения.

PTO (A/B с четырехкратным фазовым сдвигом)	
Фаза A опережает фазу B (перемещение в положительном направлении)	Фаза A следует за фазой B (перемещение в отрицательном направлении)
	
Количество импульсов	Количество импульсов

- PTO (импульс и направление (выбор направления отменен)): Если выход направления отключается в PTO (импульс и направление (выбор направления отменен)), то выход (P0) управляет импульсами. Выход P1 не используется и доступен для других целей в программе. Только команды положительного перемещения принимаются CPU в этом режиме. Управление перемещением блокирует недопустимые отрицательные конфигурации при выборе этого режима. Можно сэкономить один выход, если приложение управления перемещением работает только с одним направлением. Однофазная опция (один выход) показана на рисунке ниже (за основу взята положительная полярность):



10.3.2 Конфигурирование генератора импульсов

1. Добавление технологического объекта:
   - В дереве проекта открыть узел "Технологические объекты" и выбрать "Добавление нового объекта".
   - Выбрать значок "Ось" (переименовать при необходимости), и нажать "OK", чтобы открыть редактор конфигурации для объекта оси.

     Примечание

     Для сохранения целостности проекта после переименования ТО, следует загружать проект в CPU в рабочем состоянии STOP. Имя изменяется тогда, когда происходит удаление технологического объекта и создание нового технологического объекта с новым именем и новым номером блока данных.

   - Вызвать свойства "Выбрать PTO для управления осью" в пункте "Основные параметры" и выбрать требуемый импульс.

     Примечание

     Если PTO не был ранее сконфигурирован в свойствах CPU , то PTO конфигурируется для использования одного из встроенных выходов.

     Если используется сигнальная плата выходов, нажать кнопку "Конфигурация устройства" для вызова свойств CPU. В "Опциях импульса" в "Параметрировании" можно сконфигурировать источник выходных данных для выхода сигнальной платы.

   - Сконфигурировать оставшиеся базовые и расширенные параметры.
2. Программирование приложения: Вставить инструкцию MC_Power в блок кода.
   - Для входа оси выбрать технологический объект "Ось", который был создан и сконфигурирован.
   - Установка входа Enable в TRUE позволяет другим инструкциям управления перемещением функционировать.
   - Установка входа Enable в FALSE отменяет другие инструкции управления перемещением.

     Примечание

     Вставлять только одну инструкцию MC_Power на ось.

3. Вставить другие инструкции управления перемещением, чтобы создать требуемое движение.

Примечание

Конфигурирование генератора импульсов для выходов на сигнальной плате: Выбрать свойства "Генераторы импульсов (PTO/PWM)" для CPU (в конфигурации устройства) и активировать генератор импульсов. Для каждого S7-1200 CPU V1.0, V2.0, V2.1 и V2.2 доступно два генератора импульсов. Для S7-1200 CPU V3.0 и V4.0 доступно четыре генератора импульсов. В той же области конфигурации под "Опции импульса" выбрать для генератора импульсов применение: PTO".

Примечание

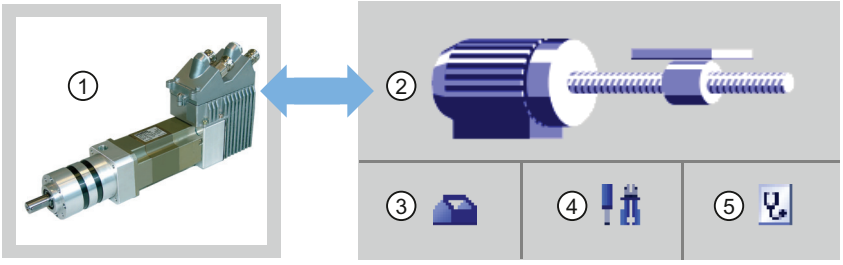
CPU обсчитывает задачи управления перемещением в "секциях" или сегментах по 10 мс. Пока одна секция выполняется, следующая секция ожидает выполнения в очереди. Если задача управления перемещением для оси прерывается (через выполнение другой новой задачи управления перемещением для оси), новая задача управления перемещением может не выполняться максимум в течение 20 мс (остаток от текущей секции плюс ожидающая в очереди секция).

10.3.3 Нерегулируемое управление перемещением

10.3.3.1 Конфигурирование оси

Нерегулируемая ось подключается к PLC и приводу через интерфейс шагового задания (PTO). Для задач управления перемещением с PTO модулю CPU нужны либо встроенные цифровые I/O, либо цифровые I/O на сигнальной плате (SB). Это ограничивает число осей, доступных для небольших PLC.

STEP 7 предлагает инструменты для конфигурирования, ввода в эксплуатацию и диагностики для TO "Ось".



1. Привод
2. Технологический объект
3. Конфигурация
4. Ввод в эксплуатацию
5. Диагностика

Примечание

Для CPU с версий прошивки до V2.2 для PTO потребуется функционального внутреннего высокоскоростного счетчика (HSC). Это означает, что использование соответствующего HSC для других целей будет невозможным.

Между PTO и HSC существует четкая взаимосвязь. Если PTO1 активируется, то выполняется подключение к HSC1. Если PTO2 активируется, то выполняется подключение к HSC2. Нельзя считывать текущее значение (напр., в ID1000) при возникновении импульсов.

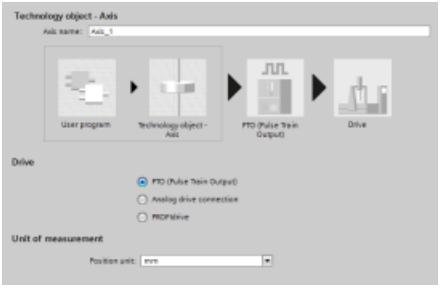
Для CPU от S7-1200 V3.0 эти ограничения отсутствуют. Все HSC остаются доступными для использования в программе, если в этом CPU сконфигурированы импульсные выходы.

Таблица 10-39 Инструменты STEP 7 для управления перемещением

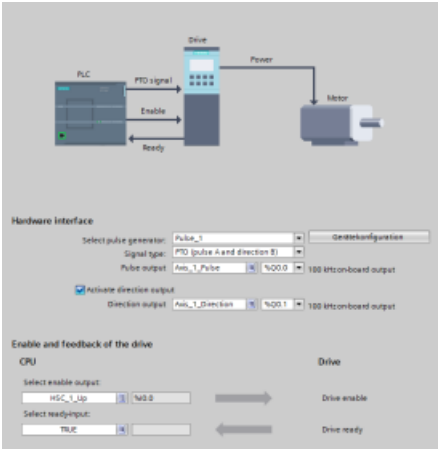
Инструмент	Описание
Конфигурация	Конфигурирует следующие свойства технологического объекта "Ось": • Выбор PTO, который необходимо использовать, и конфигурация интерфейса привода • Свойства механической части и передаточного числа привода (или машины / установки) • Свойства предельных позиционных значений, динамики и движения к референтной точке Сохранить конфигурацию в блоке данных технологического объекта.
Ввод в эксплуатацию	Тестирование функций оси без создания программы пользователя. При запуске инструмента отображается панель управления. На панели управления доступны следующие команды: • Включение и выключение оси • Перемещение оси в толчковом режиме • Абсолютное и относительное перемещение оси • Выполнение движения оси к референтной точке • Квитирование ошибок Для команд управления перемещением может указываться скорость, а также ускорение/замедление. Панель управления также показывает текущее состояние оси.
Диагностика	Мониторинг текущего состояния и информация об ошибках для оси и привода.



Древовидная структура выбора для оси PTO не включает в себя меню настроек для энкодера, модульной функции, позиционного контроля и регулирования.

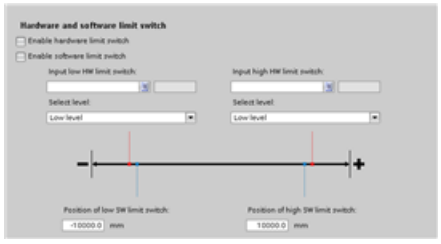


После создания технологического объекта для оси, начинается конфигурирование оси, при этом определяются основные параметры, такие как PTO и конфигурация интерфейса привода. Конфигурируются и другие свойства оси, например, предельные позиционные значения, динамика и движение к референтной точке.

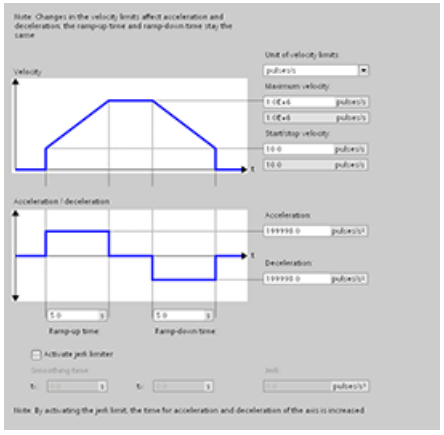


Примечание

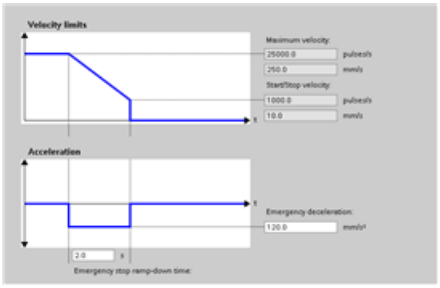
Может потребоваться адаптация значений входных параметров инструкций управления перемещением к новым единицам размерности в программе пользователя.



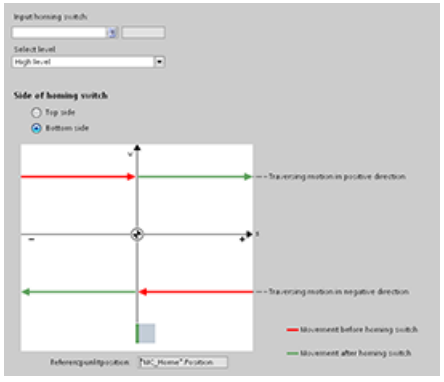
Сконфигурировать свойства для сигналов привода, механических компонентов привода и контроля положения (аппаратных и программных конечных выключателей).

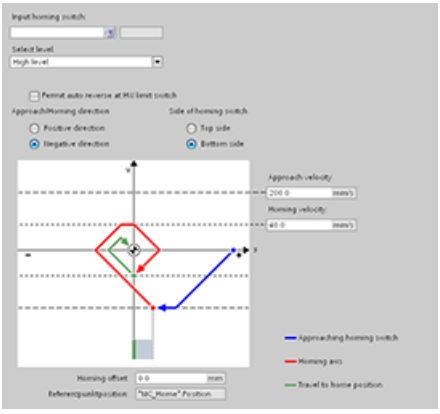


Конфигурируется динамика управления перемещением и поведение при команде аварийного отключения.



Также настраивается поведение при реферировании (пассивном и активном).





На панели управления "Ввод в эксплуатацию" можно тестировать функциональность независимо от программы пользователя.

Кликнуть по символу "Пуск", чтобы ввести ось в эксплуатацию.

Панель управления показывает текущее состояние оси. Кроме этого можно не только включить и выключить ось, но и протестировать позиционирование оси (в абсолютных и относительных значениях), а также задать скорость, ускорение и замедление. Также можно протестировать движение к референтной точке и толчковый режим. Панель управления позволяет квитировать ошибки.

10.3.3.2 Ввод в эксплуатацию

Диагностическая функция "Биты состояния и ошибок"

При помощи функции диагностики "Биты состояния и ошибок" можно осуществлять контроль за основными сообщениями о состоянии и ошибках в работе оси. Диагностический дисплей функции доступен онлайн в ручном режиме и в автоматическом режиме, когда ось активна:

Таблица 10-40 Состояние оси

Состояние	Описание
Разблокирована	Ось включена и готова к выполнению задач управления перемещением. (Переменная (тег) технологического объекта: <имя оси>.StatusBits.Enable).
Реферирование выполнено	Ось проходит реферирование и теперь может выполнять задачи абсолютного позиционирования инструкции управления перемещением "MC_MoveAbsolute". Для относительного реферирования ось может не находиться на референтной точке. Особые ситуации: • Состояние при активном движении к референтной точке (реферировании) = FALSE. • Если для оси, которая находится на референтной точке, выполняется пассивное реферирование, то состояние при пассивном реферировании устанавливается на TRUE. (Переменная (тег) технологического объекта: <имя оси>.StatusBits.HomingDone).
Ошибка	В работе технологического объекта "Ось" обнаружена ошибка. Более подробную информацию об ошибке можно найти в автоматическом управлении в параметрах ErrorID и ErrorInfo инструкций управления перемещением. В ручном режиме работы поле "Последняя ошибка" на панели управления содержит подробную информацию о причине ошибки. (Переменная (тег) технологического объекта: <имя оси>.StatusBits.Error).
Панель управления активна	Режим "Ручное управление" был включен на панели управления. Панель управления имеет приоритет управления для технологического объекта "Ось". Управлять осью из пользовательской программы невозможно. (Переменная (тег) технологического объекта: <имя оси>.StatusBits.ControlPanelActive)

Таблица 10-41 Состояние привода

Состояние	Описание
Привод готов	Привод готов к работе. (Переменная (тег) технологического объекта: <имя оси>.StatusBits.DriveReady).
Ошибка	Привод сообщил об ошибке после сбоя его сигнала готовности. (Переменная (тег) технологического объекта: <имя оси>.ErrorBits.DriveFault).

Таблица 10-42 Состояние движения оси

Состояние	Описание
Состояние покоя	Ось в состоянии покоя. (Переменная (тег) технологического объекта: <имя оси>.StatusBits.StandStill).
Ускорение	Ось разгоняется. (Переменная (тег) технологического объекта: <имя оси>.StatusBits.Acceleration).
Постоянная скорость	Ось перемещается с постоянной скоростью. (Переменная (тег) технологического объекта: <имя оси>.StatusBits.ConstantVelocity).
Замедление	Ось выполняет торможение (замедляет ход). (Переменная (тег) технологического объекта: <имя оси>.StatusBits.Deceleration).

Таблица 10-43 Состояние режима перемещения

Состояние	Описание
Позиционирование	Ось выполняет задачу позиционирования поинструкции управления перемещением "MC_MoveAbsolute" или "MC_MoveRelative" или от панели управления. (Переменная (тег) технологического объекта: <имя оси>.StatusBits.PositioningCommand).
Команда скорости	Ось выполняет задачу с установленной скоростью поинструкции управления перемещением "MC_MoveVelocity" или "MC_MoveJog" или от панели управления. (Переменная (тег) технологического объекта: <имя оси>.StatusBits.SpeedCommand).
Движение к референтной точке (реферирование)	Ось выполняет движение к референтной точке поинструкции управления перемещением "MC_Home" или от панели управления. (Переменная (тег) технологического объекта: <имя оси>.StatusBits.Homing).

Таблица 10-44 Биты ошибок

Ошибка	Описание
Минимальное программное ограничение достигнуто	Нижний программный конечный выключатель достигнут. (Переменная (тег) технологического объекта: <имя оси>.ErrorBits.SwLimitMinReached).
Минимальное программное ограничение превышено	Нижний программный конечный выключатель нарушен. (Переменная (тег) технологического объекта: <имя оси>.ErrorBits.SwLimitMinExceeded).
Максимальное программное ограничение достигнуто	Верхний программный конечный выключатель достигнут. (Переменная (тег) технологического объекта: <имя оси>.ErrorBits.SwLimitMaxReached).
Максимальное программное ограничение превышено	Верхний программный конечный выключатель нарушен. (Переменная (тег) технологического объекта: <имя оси>.ErrorBits.SwLimitMaxExceeded).
Отрицательное аппаратное ограничение	Наезд на нижний аппаратный конечный выключатель. (Переменная (тег) технологического объекта: <имя оси>.ErrorBits.HwLimitMin).

Ошибка	Описание
Положительное аппаратное ограничение	Наезд на верхний аппаратный конечный выключатель. (Переменная (тег) технологического объекта: <имя оси>.ErrorBits.HwLimitMax).
PTO уже используется	Вторая ось использует тот же PTO и деблокирована с помощью "MC_Power". (Переменная (тег) технологического объекта: <имя оси>.ErrorBits.HwUsed).
Ошибка конфигурации	Технологический объект "Ось" настроен неправильно, или редактируемые данные конфигурации были ошибочно изменены при работе пользовательской программы. (Переменная (тег) технологического объекта: <имя оси>.ErrorBits.ConfigFault).
Общая ошибка	Произошла внутренняя ошибка. (Переменная (тег) технологического объекта: <имя оси>.ErrorBits.SystemFault).

Функции диагностики "Состояние перемещения"

При помощи функции диагностики "Состояние перемещения" контролируется состояние движения оси. Описание соответствующего состояния перемещения представлено в следующей таблице:

Таблица 10-45 Состояние перемещения

Состояние	Описание
Фактическая позиция	В поле "Фактическая позиция" указывается измеренная позиция оси. Если ось не реферирована, то это значение является позиционным значением относительно позиции включения оси. (Переменная (тег) технологического объекта: <имя оси>.ActualPosition)
Фактическая скорость	В поле "Фактическая скорость" указывается измеренная скорость оси. (Переменная (тег) технологического объекта: <имя оси>.ActualVelocity)
Заданное значение положения	В поле "Заданное значение положения" указывается измеренное заданное значение положения оси. Если ось не реферирована, то это значение является позиционным значением относительно позиции включения оси. (Переменная (тег) технологического объекта: <имя оси>.Position)
Заданное значение скорости	В поле "Заданное значение скорости" указывается вычисленное заданное значение скорости оси. (Переменная (тег) технологического объекта: <имя оси>.Velocity)
Заданная позиция	В поле "Заданная позиция" указывается текущая целевая позиция активной команды позиционирования или таблицы команд оси. Значение заданной позиции действительно только во время выполнения команды позиционирования. (Переменная (тег) технологического объекта: <имя оси>.StatusPositioning.TargetPosition)
Оставшаяся длина перемещения	В поле "Оставшаяся длина перемещения" указывается оставшаяся длина перемещения для активной команды позиционирования или таблицы команд оси. Значение "Оставшаяся длина перемещения" действительно только во время выполнения команды позиционирования. (Переменная (тег) технологического объекта: <имя оси>.StatusPositioning.Distance)

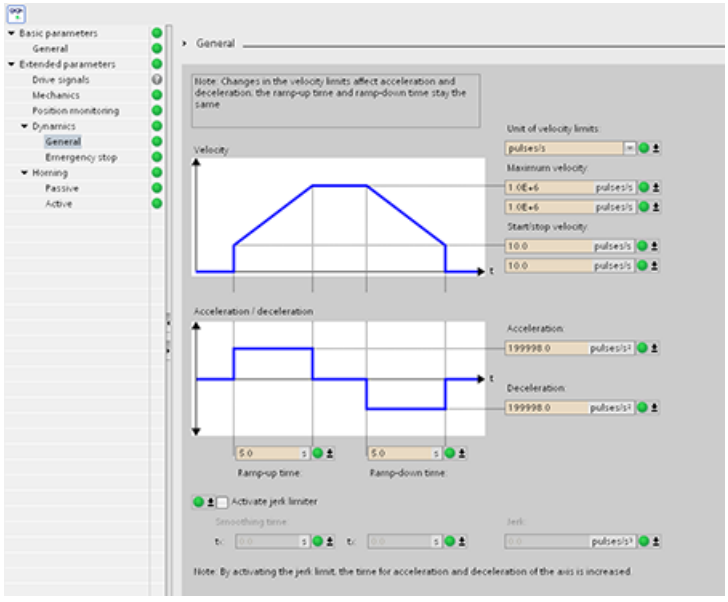
Таблица 10-46 Динамические ограничения

Динамическое ограничение	Описание
Скорость	В поле "Скорость" отображается сконфигурированная максимальная скорость оси. (Переменная (тег) технологического объекта: <имя оси>.Config.DynamicLimits.MaxVelocity).
Ускорение	В поле "Ускорение" отображается текущее сконфигурированное ускорение оси. (Переменная (тег) технологического объекта: <имя оси>.Config.DynamicDefaults.Acceleration).
Замедление	В поле "Замедление" отображается текущее сконфигурированное замедление оси. (Переменная (тег) технологического объекта: <имя оси>.Config.DynamicDefaults.Deceleration).

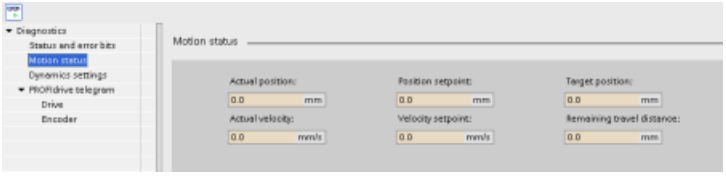
Управление стартовыми значениями при управлении перемещением

Можно отредактировать фактические значения параметров конфигурации управления перемещением так, чтобы поведение процесса могло быть оптимизировано в режиме онлайн.

Открыть "Технологические объекты" для управления перемещением и их объект "Конфигурация". Чтобы получить доступ к управлению стартовыми значениями, кликнуть по "значку очков" в левом верхнем углу диалогового окна:



Следует использовать функцию диагностики "Состояние перемещения" для мониторинга состояния перемещения оси. Диагностический дисплей функции доступен онлайн в ручном режиме и в автоматическом режиме, когда ось активна:



Теперь можно изменить значение любого из параметров конфигурации управления перемещением, как показано на рисунке ниже.

Можно сравнить фактическое значение со стартовым значение в проекте (офлайн) и стартовым значением PLC (онлайн) для каждого параметра. Это необходимо, чтобы выявить онлайн/офлайн различия блока данных технологического объекта (TO-DB) и знать значения, которые будут использоваться в качестве текущих значений при следующем переходе PLC из STOP в START. Кроме того, значок сравнения предоставляет визуальную индикацию для упрощения обнаружения различий в онлайн/офлайне:

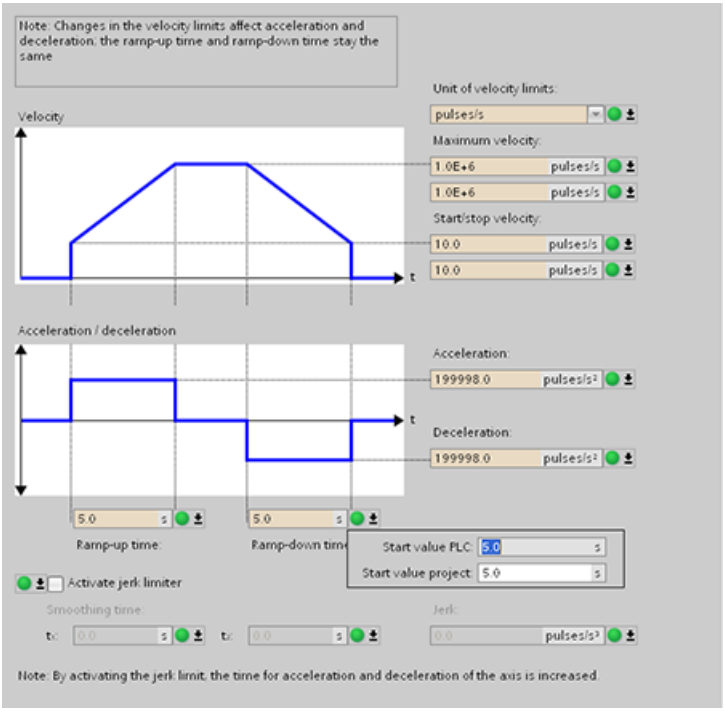
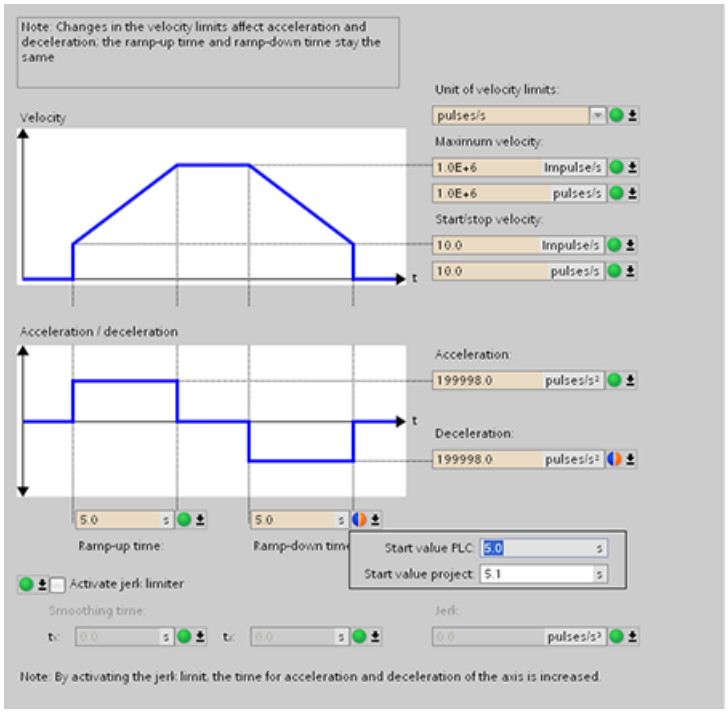


Рисунок выше представляет экран параметров управления перемещением со значками сравнения. Символы показывают, какие из значений различаются в онлайн и офлайн проектах. Зеленый значок обозначает одинаковые значения. Синий/оранжевый значок указывает на то, что значения отличаются друг от друга.

Дополнительно можно нажать кнопку параметра с направленной вниз стрелкой, чтобы открыть небольшое окно со стартовым значением проекта (офлайн) и стартовым значением PLC (онлайн) для каждого параметра.



10.3.4 Регулируемое управление перемещением

10.3.4.1 Конфигурирование оси

Регулируемая ось подключается к PLC и приводу с использованием одного из двух соединений:

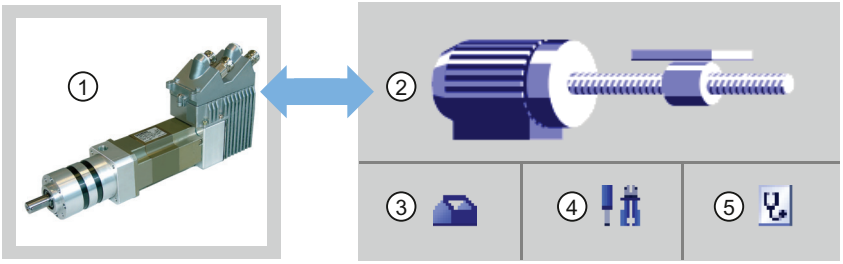
- Аналоговый привод (интерфейс аналогового задания): Для такого подключения могут использоваться встроенные аналоговые I/O или аналоговые I/O на SB или на сигнальном модуле (SM). PTO не используются. Для приложения доступно следующее разрешение аналоговых I/O:
  - Встроенные I/O: 10 бит (самое низкое разрешение I/O)
  - I/O сигнальные платы (SB): 12 бит
  - I/O на SM: 14 бит (самое высокое разрешение I/O)
- PROFIdrive: Здесь речь идет о сетевом решении, PTO не используются.

Для регулируемой оси также нужен энкодер. Энкодеры могут подключаться к следующим элементам:

- Интерфейс энкодера на приводе
- Высокоскоростные счетчики (HSC)
- Технологические модули (TM)
- Энкодер PROFIdrive к PROFINET / PROFIBUS

Соединение PROFIdrive или интерфейс аналогового задания поддерживают максимум восемь приводов (или осей).

STEP 7 предлагает инструменты для конфигурирования, ввода в эксплуатацию и диагностики для ТО "Ось".



① Привод
② Технологический объект
③ Конфигурация
④ Ввод в эксплуатацию
⑤ Диагностика

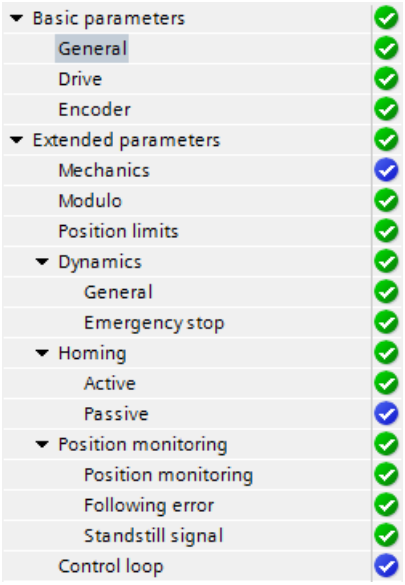
Таблица 10-47 Инструменты STEP 7 для регулируемого управления перемещением

Инструмент	Описание
Конфигурация	Конфигурирует следующие свойства технологического объекта "Ось": • Выбор требуемого интерфейса аналогового задания или выбор PROFIdrive и конфигурирование интерфейса привода и энкодера • Свойства механической части и передаточного числа привода и энкодера (или машины / установки) • Свойства предельных позиционных значений, динамики и движения к референтной точке Сохранить конфигурацию в блоке данных технологического объекта.
Ввод в эксплуатацию	Тестирование функций оси без создания программы пользователя. При запуске инструмента отображается панель управления. На панели управления доступны следующие команды: • Включение и выключение оси • Перемещение оси в толчковом режиме • Абсолютное и относительное перемещение оси • Выполнение движения оси к референтной точке • Квитирование ошибок Для команд управления перемещением может указываться скорость, а также ускорение/замедление. Панель управления также показывает текущее состояние оси.
Диагностика	Мониторинг текущего состояния и информация об ошибках для оси и привода.

Примечание

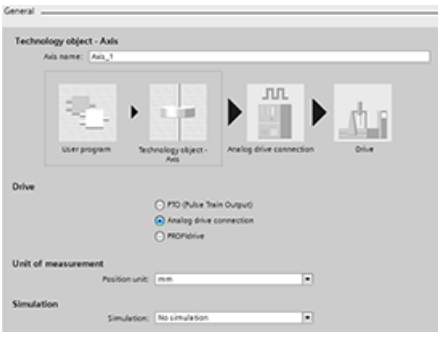
Может потребоваться адаптация значений входных параметров инструкций управления перемещением к новым единицам размерности в программе пользователя.

После создания технологического объекта для оси, начинается конфигурирование оси, при этом определяются основные параметры, выбирается интерфейс аналогового задания или PROFIdrive, а также конфигурация привода и энкодера.



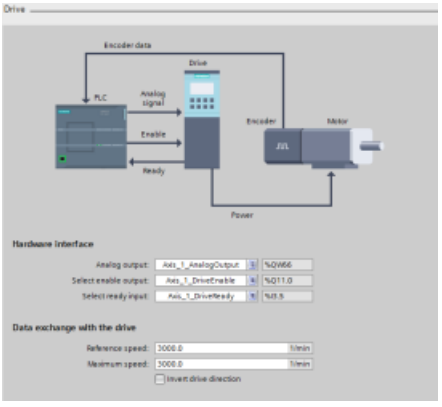
Древовидная структура выбора для интерфейса аналогового задания или PROFIdrive включает в себя меню настроек для энкодера, модульной функции, позиционного контроля и регулирования.

Конфигурация интерфейса аналогового задания



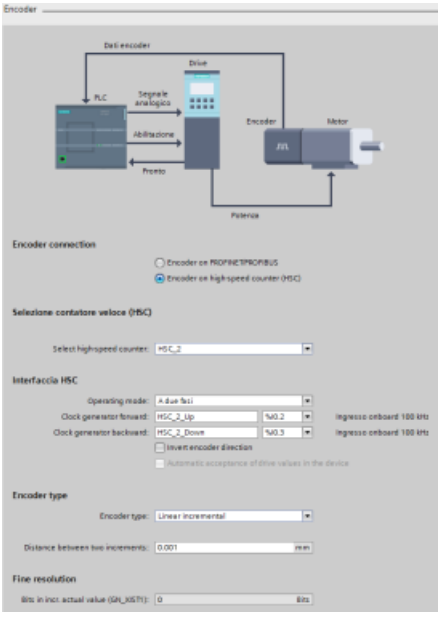
В диалоговом окне конфигурирования "Общее" выбираются следующие параметры:

- Радиокнопка "Интерфейс аналогового задания"
- Единица измерения



В диалоговом окне конфигурирования "Привод" выбираются следующие параметры:

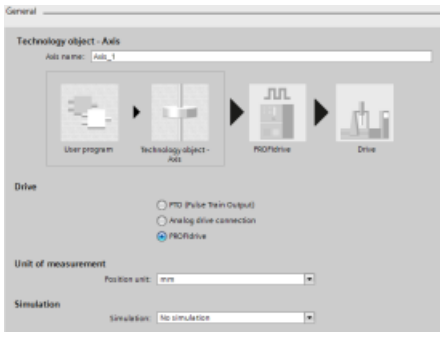
- Аппаратные выходы аналогового привода
- Скорости обмена данными с приводом
  Примечание: Максимальная скорость должна быть выше или равна опорной скорости (номинальной скорости).



В диалоговом окне конфигурирования "Энкодер" выбираются следующие параметры:

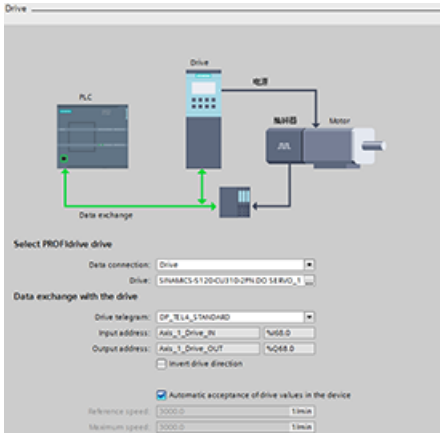
- Подключение энкодера аналогового привода (напр., высокоскоростной счётчик (HSC))
- Интерфейс HSC
- Тип энкодера
- Точное разрешение

Конфигурация PROFIdrive



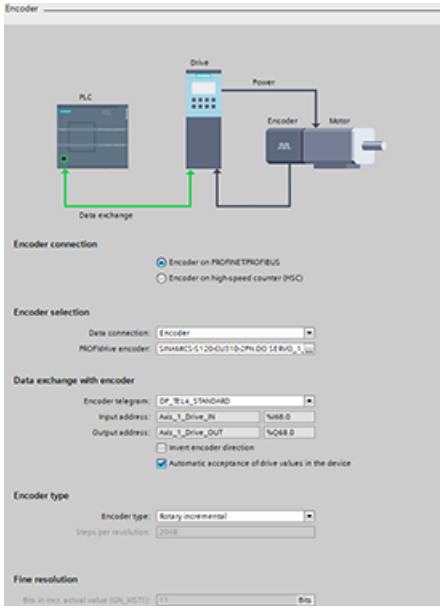
В диалоговом окне конфигурирования "Общее" выбираются следующие параметры:

- Радиокнопка "PROFIdrive"
- Единица измерения



В диалоговом окне конфигурирования "Привод" выбираются следующие параметры:

- Привод PROFIdrive
- Обмена данными с приводом
  Примечание: Максимальная скорость должна быть выше или равна опорной скорости (номинальной скорости).



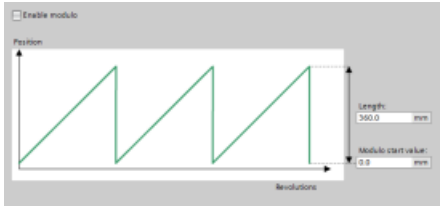
В диалоговом окне конфигурирования "Энкодер" выбираются следующие параметры:

- Подключение энкодера PROFIdrive (напр., энкодера PROFIdrive на PROFINET)
- Энкодер PROFIdrive
- Обмена данными с энкодером
- Тип энкодера
- Точное разрешение

Расширенные параметры

Кроме этого, могут быть определены следующие свойства регулируемой оси:

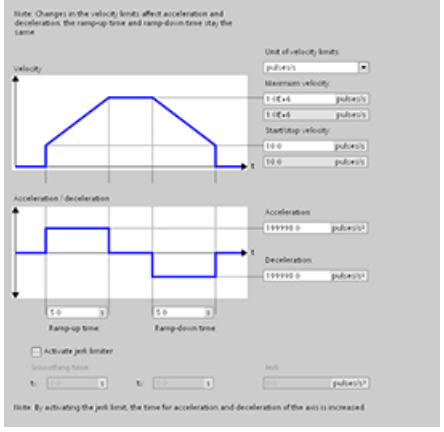
- Модульная функция
- Границы позиции
- Динамика
- Движение к референтной точке (реферирование)
- Позиционный контроль
- Ошибка рассогласования
- Сигнал состояния покоя
- Регулирование



Модульная функция: Можно настроить "модульную" ось для перемещения нагрузки в замкнутой области, которая имеет стартовое значение / стартовую позицию и заданную длину. Если позиция нагрузки достигает конца этой области, она автоматически снова устанавливается на стартовое значение. После установки флажка "Активировать модульную функцию" включаются поля "Длина" и "Стартовое значение модульной функции".



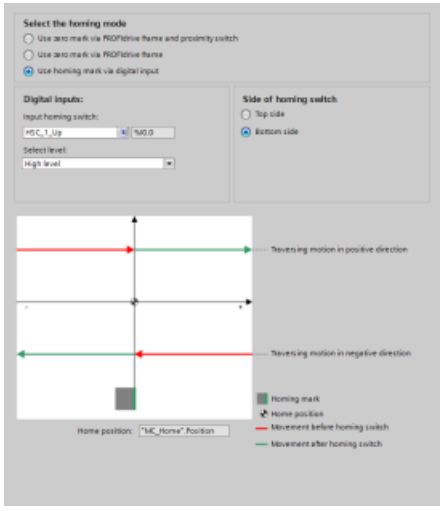
Границы позиции: Можно сконфигурировать свойства для сигналов привода, механических компонентов привода и контроля положения (аппаратных и программных конечных выключателей).

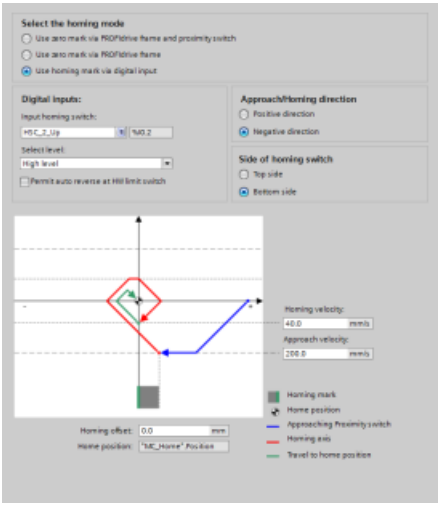


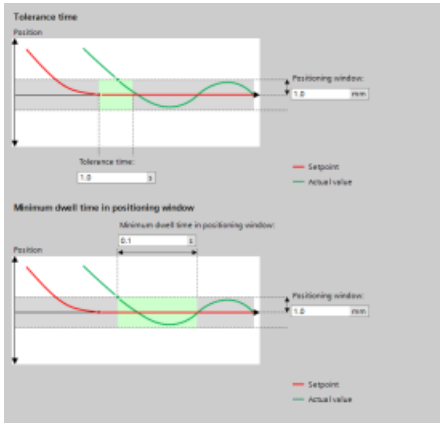
Динамика: Конфигурируется динамика управления перемещением и поведение при команде аварийного отключения.

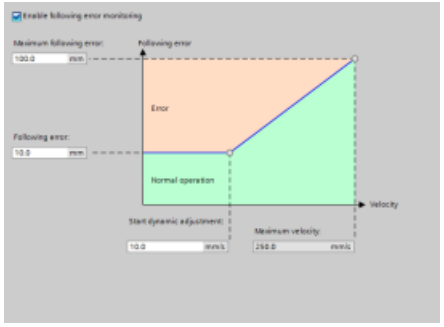


Движение к референтной точке (реферирование): Также настраивается поведение при реферировании (пассивном и активном).









"Позиционный контроль": Можно сконфигурировать хронометрический допуск, а также минимальное время ожидания для окна позиционирования.

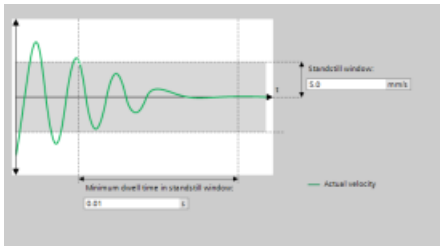
Система напрямую соединяет три следующих параметра с TO-DB оси:

- Окно позиционирования
- Хронометрический допуск
- Наименьшее время ожидания в окне позиционирования

Примечание: Минимальное значение для поля "Окно позиционирования" составляет "0,001", максимальное значение = "1 E+12".

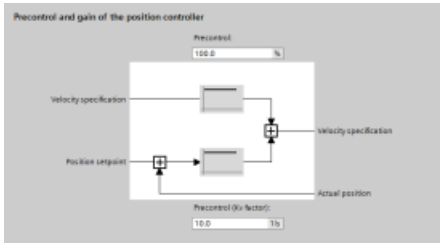
"Ошибка рассогласования": Можно настроить разницу допустимого интервала ошибок в диапазоне скоростей. Для активации ошибки рассогласования, установить флажок "Включить контроль ошибки рассогласования". Можно определить следующие параметры:

- Максимальная ошибка рассогласования
- Ошибка рассогласования
- Начало динамической адаптации
- Максимальная скорость



"Сигнал состояния покоя": Можно определить следующие параметры:

- Наименьшее время ожидания в окне состояния покоя
- Окно состояния покоя.



"Регулирование": Можно настроить коэффициент усиления по скорости "Предусиление (коэффициент Kv)".

На панели управления "Ввод в эксплуатацию" можно тестировать функциональность независимо от программы пользователя.

Кликнуть по символу "Пуск", чтобы ввести ось в эксплуатацию.

Панель управления показывает текущее состояние оси. Кроме этого можно не только включить и выключить ось, но и протестировать позиционирование оси (в абсолютных и относительных значениях), а также задать скорость, ускорение и замедление. Также можно протестировать движение к референтной точке и толчковый режим. Панель управления позволяет квитировать ошибки.

10.3.4.2 ServoOB

При создании технологического объекта для S7-1200 управления перемещением, CPU автоматически создает организационный блок "MC-Servo (OB 91)" для обработки технологических объектов. Функциональность управления перемещением технологических объектов создает собственный класс приоритетов и система выполнения SIMATIC S7-1200 вызывает OB согласно прикладному циклу.

MC-Servo OB имеет защиту от записи. Изменение содержания невозможно.

Алгоритмы для управления позиционированием всех сконфигурированных в CPU для управления перемещением TO вычисляются в OB MC-Servo.

В соответствии с требованиями к качеству управления и нагрузки на систему можно настроить прикладной цикл и приоритет организационного блока. Несколько ServoOB могут быть связаны с событием "CyclicServoEvent". На странице свойств OB MC-Servo предлагается два редактируемых значения для события CyclicServoEvent:

- Номер приоритета
- Время цикла

Организационные блоки OB MC-PreServo и OB MC-PostServo являются экземплярами ServoOB и имеют тип ServoOB. Эти OB являются необязательными, могут существовать только при наличии OB MC-Servo и содержат код пользователя. Все три OB (MC-PreServo, MC-Servo и MC-PostServo) должны обрабатываться на одном уровне выполнения и

запускаются одним событием CyclicServoEvent. Прошивка PLC обрабатывает OB в последовательности их номеров блоков:

OB-экземпляр	OB-номер	Событие объектной модели системы автоматизации (ASOM)	Событие объектной модели системы проектирования (ESOM)
MC-PreServo	67	ServoOB	-
MC-Servo	91	ServoOB	ServoOB
MC-PostServo	95	ServoOB	-

Если сравнивать с ASOM, то объектная модель системы проектирования (ESOM) связывает событие только с OB MC-Servo. ESOM не связывает событие с OB MC-PreServo или с OB MC-PostServo.

Конфигурирование оси с MC-Servo (OB 91) для прикладного цикла

MC-Servo (OB 91) для прикладного цикла

Прикладной цикл, в котором вызывается OB MC-Servo, может быть определен в свойствах организационного блока:

- Синхронно с ШИНОЙ: OB MC-Servo вызывается синхронно с системной шиной. Так передачи определяется в свойствах выбранной системной шины.
- Циклически: OB MC-Servo вызывается циклически с назначенным прикладным циклом.

Для исключения прерываний при выполнении программы в CPU, прикладной цикл определяется в зависимости от числа используемых осей следующим образом:

Прикладной цикл = Число осей x 2 мс

Число осей	Прикладной цикл
1	2 мс
2	4 мс
4	8 мс
8	16 мс

Привод SINAMICS G120 обновляет образ процесса привода каждые 4 мс. Для улучшения управления, следует установить прикладной цикл MC-Servo (OB 91) равным 4 мс или кратным 4 мс.

Переполнение MC-Servo (OB 91)

Выбранный прикладной цикл должен быть достаточно длинным, чтобы все технологические объекты для управления перемещением могли быть обработаны в одном цикле. Если прикладной цикл не выдерживается, то происходит переполнение.

При переполнении MC-Servo (OB 91) модуль CPU не переходит в состояние STOP. (Информация в системе интерактивной помощи TIA Portal касательно режима STOP при переполнении MC-Servo (OB 91) не соответствует действительности.)

При необходимости в случае переполнения MC-Servo (OB 91) модуль CPU может быть переведен в состояние STOP с помощью OB прерывания по ошибке времени (OB 80).

Раздел образа процесса "OB-Servo-PIP"

Для оптимального управления всеми используемыми управлением перемещением модулями ввода-вывода (например, аппаратными конечными выключателями), следует назначить их разделу образа процесса "OB-Servo-PIP". После назначения модули ввода-вывода будут обрабатываться одновременно с технологическим объектом.

Если управление перемещением использует высокоскоростной счетчик (HSC), управление перемещением автоматически назначает HSC разделу образа процесса "OB-Servo-PIP".

10.3.4.3 Режим управления по скорости

Режим управления по скорости позволяет выполнять перемещение позиционирующей оси в режиме работы "Управление по скорости". С помощью инструкции MC_Power ось может быть активирована. С помощью инструкций MC_MoveVelocity и MC_MoveJog ось может двигаться по заданным значениям скорости. На движение оси не влияет ни ошибка датчика, ни отсутствие действительных значений. Режим работы "Управление по скорости" устанавливает следующие ограничения:

- Позиционный регулятор оси отключается
- Заданные значения скорости для привода устанавливаются напрямую
- Заданное значение для позиции оси устанавливается на ноль
- Фактическое положение нижнего предельного значения обновляется при правильных значениях датчика
- Ошибка рассогласования и ошибка контроллера не определяются, и соответствующие значения устанавливаются на ноль

Для включения и выключения режима работы "Управление по скорости" могут использоваться следующие три инструкции и эти параметры:

- MC_Power.StartMode (Int)
- MC_MoveVelocity.PositionControlled (Bool)
- MC_MoveJog.PositionControlled (Bool)

MC_Power

С помощью инструкции MC_Power ось может быть активирована в режиме работы "Управление по скорости". Такая возможность существует, если отсутствуют правильные значения, или если ось не может перейти в режим работы "Управление по положению".

Могут использоваться только стартовые режимы "0" и "1". Другие значения возвращают ошибку:

StartMode	PTO ось	Сервоось
0	Игнорировать	Управления по скорости
1	Игнорировать	Управление по положению
Прочее	Недействительный режим работы MC_Power.ErrorID = 0x8412 MC_Power.ErrorID = 0x0011	Недействительный режим работы MC_Power.ErrorID = 0x8412 MC_Power.ErrorID = 0x0011

При выключении ТО "Ось" через "MC_Power.Enable = FALSE" режим работы сохраняется. Поведение оси зависит от режима работы:

StartMode	Режим работы: Управление по положению	Режим работы: Управления по скорости
0: Аварийное выключение	Ось выполняет торможение с управлением по положению с "DynamicDefaults. EmergencyDeceleration" до базовых значений ActualPosition и ActualVelocity.	Ось выполняет торможение с управлением по положению с "DynamicDefaults. EmergencyDeceleration" до базовых значений ActualVelocity, если таковые доступны.
1: Немедленный останов	Привод останавливается по рампе "ВЫКЛ3" в подключенном приводе.	Привод останавливается по рампе "ВЫКЛ3" в подключенном приводе.
2: Аварийный останов с управлением рывком	Торможение осей с установленным аварийным замедлением на основе заданной позиции с управлением по положению. Если управление рывком активировано, то учитывается сконфигурированный рывок.	Торможение осей с установленным аварийным замедлением на основе заданной скорости с управлением по скорости. Если управление рывком активировано, то учитывается сконфигурированный рывок.

При остановке привода режим работы не может быть изменен, так как новые команды движения будут приняты только после перехода привода в состояние покоя. После перехода привода в состояние покоя, ось снова может быть активирована.

MC_MoveVelocity/MC_MoveJog

Независимо от режима работы (управление по скорости/положению), блоки изменяют нижнее предельное значение в зависимости от сконфигурированного рабочего состояния. Это возможно при регулируемом управлении перемещением или в состоянии покоя.

MC_Halt

Инструкция MC_Halt не изменяет режим работы. В режиме работы "Управление по скорости" вычисленная рампа торможения зависит от заданной скорости для сконфигурированного замедления.

По достижении окна состояния покоя, команда завершается и сигнализируется "DONE = TRUE".

Прочие команды управления перемещением

Режим работы "Управление по скорости" остается активным, пока активна одна из следующих команд:

- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity (PositionControlled = TRUE)
- MC_MoveJog (PositionControlled = TRUE)
- MC_Home:
  - Активное движение к референтной точке (режим работы 3)
  - Другие режимы работы (пассивный, прямой, настройка абсолютного энкодера) отклоняются с ErrorID:
    ErrorId 8207: "Command rejected" / "Задание отклонено"
    ErrorInfo 006B: "Вызов в режиме работы "Управления по скорости" недопустим".

TO-DB оси

TO-DB оси показывает режим работы, если одна из трех инструкций управления перемещением (MC_Power, MC_MoveVelocity или MC_MoveJog) выполняется в режиме работы "Управления по скорости":

- Axis.Statusbit.NonPositionControlled = TRUE
- Axis.Position = 0,0

Расчет новой заданной позиции

После перехода из режима работы "Управления по скорости" в "Регулируемое управление перемещением", заданная позиция должна быть рассчитана заново:

- С предуправлением (kpc > 0): Позиция = ActualPosition + ActualVelocity * vtc
- Без предуправления (kpc = 0): Позиция = ActualPosition + ActualVelocity / kv (vtc = постоянная времени предуправления; kv = усиление позиционного регулятора)

Программные конечные выключатели

В режиме управления по скорости программные конечные выключатели отключены.

Аппаратные конечные выключатели

В режиме управления по скорости аппаратные конечные выключатели поддерживаются.

10.3.4.4 Поддержка телеграммы 4

PROFdrive телеграмма 4 содержит значения привода и два значения различных энкодеров. Первое значение энкодера поступает от энкодера на двигателе. Второе значение энкодера поступает от дополнительного внешнего энкодера.

Внешний энкодер подключен напрямую к управляющему модулю SINAMICS, и он передает оба значения в телеграмме 4.

Диалоговое окно конфигурирования "Привод"

Телеграмма 4 может быть настроена в конфигурации оборудования и в диалоговом окне конфигурирования "Привод".

Диалоговое окно конфигурирования "Энкодер"

В диалоговом окне конфигурирования "Энкодер" предлагается две опции:

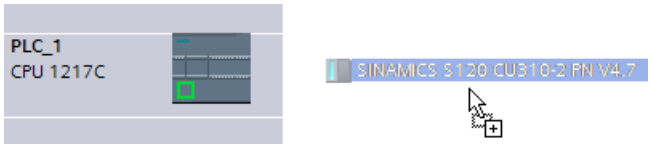
- Энкодер на PROFINET/PROFIBUS
- Энкодер на высокоскоростном счетчике (HSC)

"Энкодер на PROFINET/PROFIBUS" является выбором по умолчанию, но доступны обе опции.

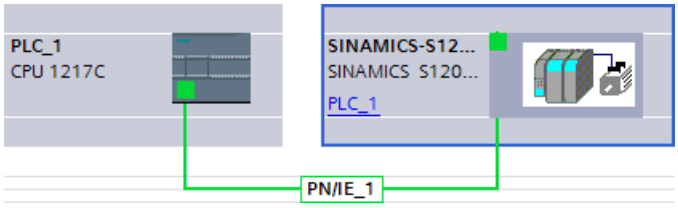
Пример: Конфигурирование SINAMICS S120 и энкодера с телеграммой 4

1. Выбор преобразователя частоты SINAMICS:
   Следует использовать каталог оборудования для добавления преобразователя частоты SINAMICS S120 CU310-2 PN V4.7. Для этого раскрыть следующие контейнеры:
   - Прочие полевые устройства
   - PROFINET IO
   - Приводы
   - SIEMENS AG
   - SINAMICS

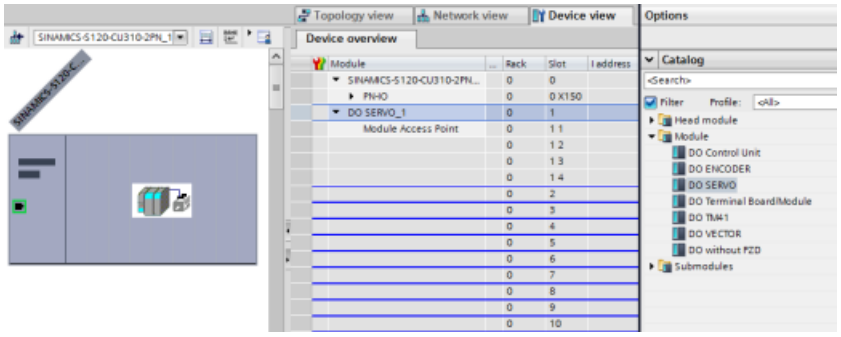
   Добавить привод так, как показано на рисунках ниже:



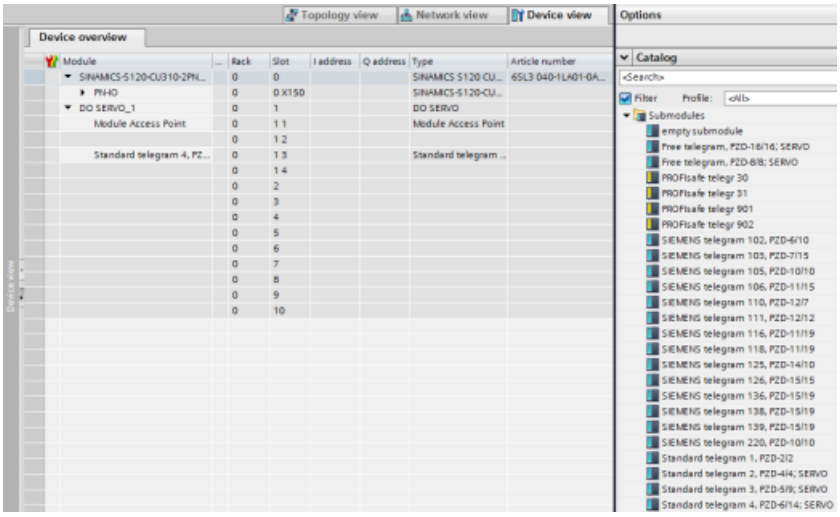
   Сконфигурировать сеть PROFINET.



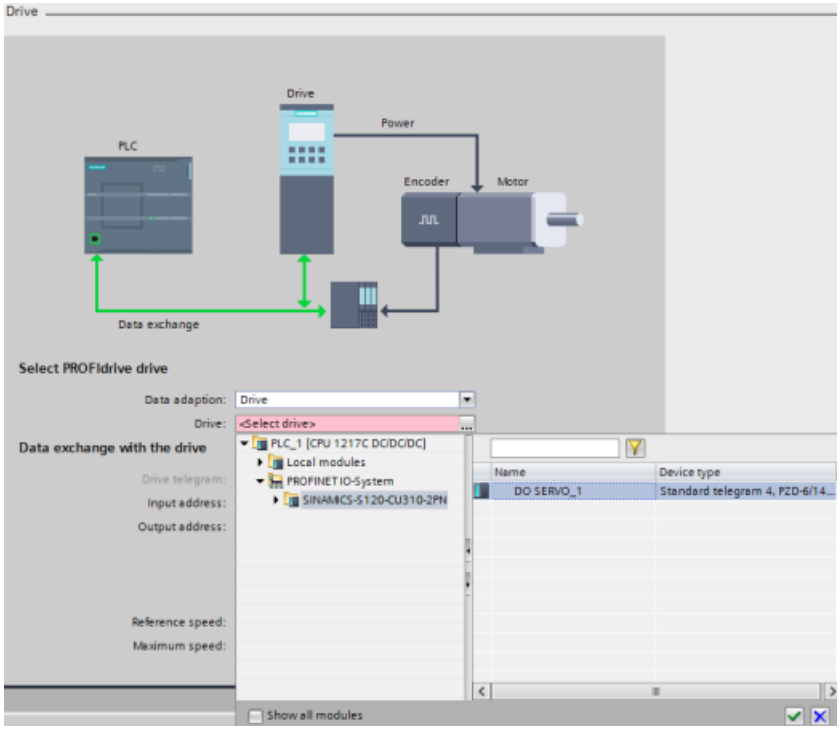
2. Выбор приводного объекта DO SERVO:
   - Двойной клик на вкладке сети по преобразователю частоты SINAMICS S120 CU310-2 PN V4.7.
   - Открыть обзор устройств.
   - Раскрыть в каталоге оборудования контейнер для модулей:
   - Выполнить двойной клик или перетащить приводной объект DO SERVO и вставить его в первый свободный ряд:



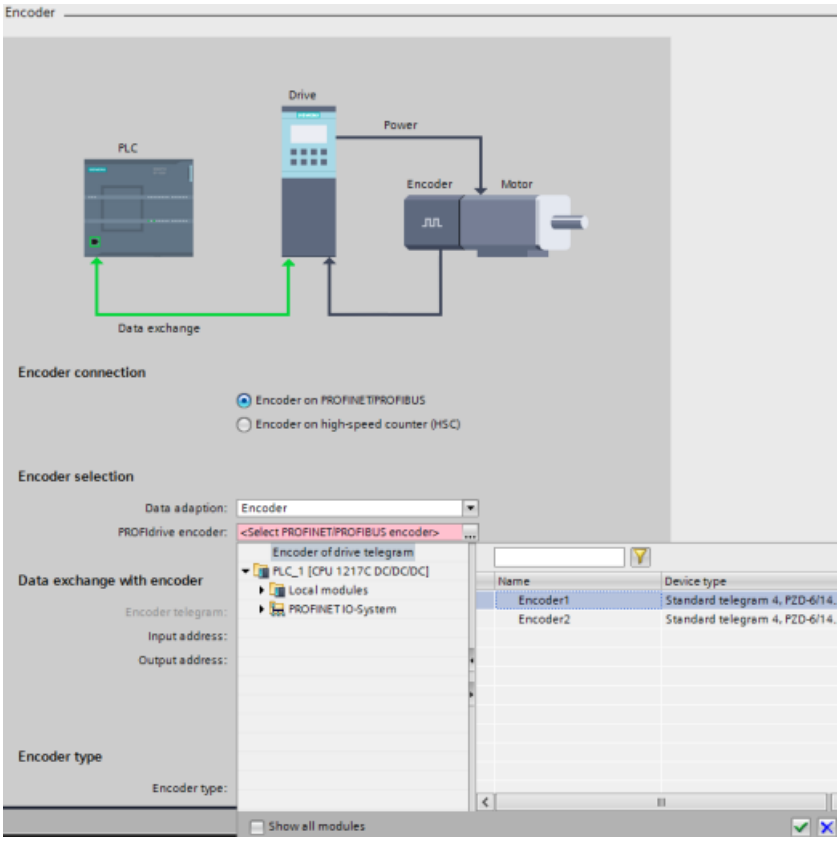
3. Выбор телеграммы 4:
   - Раскрыть в каталоге оборудования контейнер для субмодулей:
   - Выполнить двойной клик или перетащить "Стандартная телеграмма 4, PZD-6/15;SERVO" и вставить ее во второй свободный ряд:
   - Следует пропустить одну пустую строку, чтобы вставить телеграмму 4 как показано на рисунке ниже:



4. Выбор энкодера в диалоговом окне конфигурирования "Привод":
   - Перейти в диалоговое окно конфигурирования для осей и к основным параметрам в диалоговом окне конфигурирования "Привод".
   - Перейти к полю "Выбор привода PROFIdrive", "Привод:".
   - Нажать кнопку с многоточием.
   - Двойной клик по "Система PROFINET IO", чтобы открыть эту папку.
   - Кликнуть по "SINAMICS-S120-CU310-2PN".
   - В правом окне отобразится "DO SERVO_1": "Стандартная телеграмма 4, PZD-6/14; SERVO".
   - Нажать кнопку с зеленой галочкой, чтобы ввести конфигурацию.



5. Конфигурирование энкодера:
   - После выбора телеграммы 4 в диалоговом окне конфигурирования "Привод", диалоговое окно конфигурирования "Энкодер" предлагает новый элемент "Энкодер телеграммы привода" в дереве переходов. После выбора элемента "Энкодер телеграммы привода", в правом окне появляются две записи со значениями энкодера: Энкодер 1 и энкодер 2.
   - Управление перемещением назначает один энкодер в качестве энкодера привода, а другой - в качестве внешнего энкодера. Путем выбора энкодера 1 или энкодера 2 определяется, какой энкодер станет энкодером привода. Энкодер, который не будет выбран, будет назначен внешним энкодером.
   - Энкодер привода выбран. Нажать кнопку с зеленой галочкой, чтобы ввести конфигурацию:



10.3.4.5 Моделируемая ось

Режим моделирования может быть выбран тогда, когда требуется использовать PROFIdrive или интерфейс аналогового задания на PLC без подключенного привода.

Режим моделирования потребуется для выполнения одной из следующих задач:

- Ввод в эксплуатацию запрограммированной последовательности без привода
- Тестирование программы пользователя без движения оси
- Симуляция поведения оси с использованием модели процесса без реального движения
- Привод и энкодер могут не подключаться: Тестирование возможно и без спроектированного привода; привод добавляет и конфигурируется позднее.

Конфигурирование режима моделирования

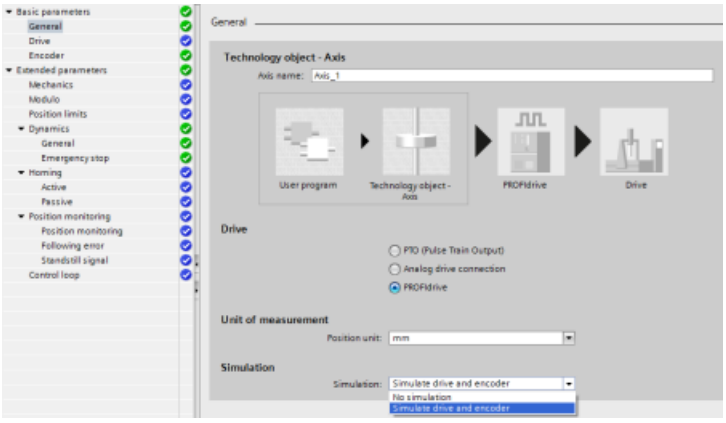
Доступны следующие опции:

Значение TO-DB	Диалоговые записи	Комментарий
0	Без моделирования	Без моделирования
1	Моделирование привода и энкодера	Моделирование с и без правильно сконфигурированных адресов оси и без сконфигурированных I/O в конфигурации оборудования

Режим моделирования поддерживается только для сервооси, поэтому можно выбрать только "PROFIdrive" и "интерфейс аналогового задания". Для PTO оси режим моделирования "0".

При переключении оси на PTO дальнейшая реализация режима моделирования должна продолжаться в TIA Portal. Кроме этого, при "Передаче данных" в случае PTO необходимо убедиться, что режим моделирования установлен на "0". Если нет, такая установка выполняется автоматически.

Оператор выбирает, как представлено на рисунке ниже, различные типы для режима моделирования в диалоговом окне "Основные параметры" > Область "Общее" > Поле "Моделирование":



Режим работы 1: "Моделирование привода и энкодера":

- В этом режиме ось работает в программе на PLC без реально подключенного оборудования (привода PROFIdrive и энкодера).
- Логические адреса I/O в TO-DB не нужны. Т.е. не нужно настраивать конфигурацию энкодера и привода, а также дополнительные цифровые сигналы (аппаратные конечные выключатели и переключатель референтной точки) в TO и в конфигурации оборудования. Ось не отправляет данные на логические адреса энкодера и привода.
- Происходит симуляция TO-DB ActualSpeed.
- PROFIdrive не использует телеграмму PROFIdrive.
- PLC сигнализирует для PROFIdrive диагностическую ошибку по отсутствию сконфигурированного оборудования (периферийных устройств или приводов), но это не влияет на возможности по использованию оси.

- Создание отчетов о реферировании напрямую во всех режимах работы для реферирования. Соответствующая установка позиций и статуса. Аппаратные входы не определяются.
- Все соединения могут быть пустыми.
- Панель управления осями / настроечная панель работают как обычно.
- Управляемая по скорости ось поддерживается.
- Виртуальная ось (при определении через SMC) не поддерживается, но можно использовать симулированную ось без аппаратных соединений как виртуальную ось.

Загрузка сценария:

Ось с изменениями может быть загружена в режиме моделирования при нахождении PLC в RUN. В этом случае PLC устанавливает бит состояния "RestartRequired". После перезапуска оси PLC передает изменения в рабочую память. В следующей таблице представлены все параметры оси (Actor, Sensor и PositionLimits), связанные с конфигурацией оборудования и являющиеся обязательными или опциональными для моделирования.

TO-DB параметр	Режим работы 1: Моделирование PROFIdrive
Actor.Interface.AddressIn	Опция
Actor.Interface.AddressOut	Опция
Actor.Interface.EnableDriveOutput	Опция
Actor.Interface.DriveReadyInput	Опция
Sensor.Interface.AddressIn	Опция
Sensor.Interface.AddressOut	Опция
Sensor.ActiveHoming.DigitalInputAddress	Опция
Sensor.PassiveHoming.DigitalInputAddress	Опция
PositionLimits_HW.MinSwitchAddress	Опция
PositionLimits_HW.MaxSwitchAddress	Опция

10.3.4.6 Перенос данных

Обзор

Предлагается возможность для переноса значений привода и энкодера, которые могут быть считаны из модуля привода и энкодера и конфигурация которых совпадает как в PLC, так и в приводе/энкодере.

Конфигурирование RT переноса данных

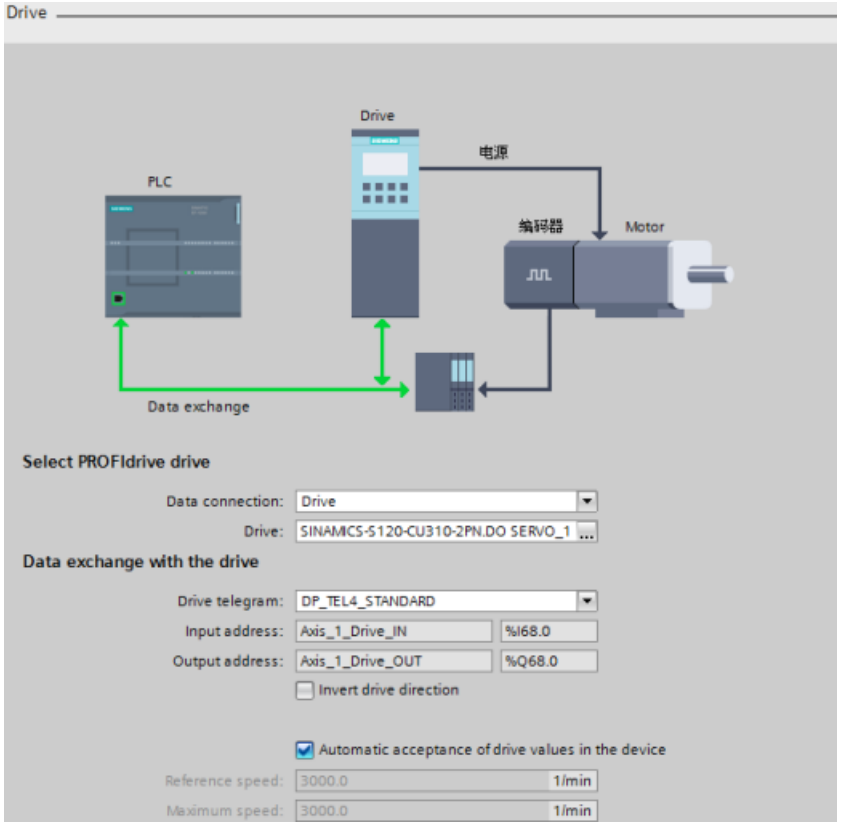
CPU конфигурирует RT переноса данных в TO-DB для соответствующего привода и энкодера:

- <axis>.Sensor[i].DataAdaptation: DINT [0:NO | 1:YES]
- <axis>.Actor.DataAdaptation: DINT [0:NO | 1:YES]

Можно изменить конфигурацию RT переноса данных в режиме RUN с помощью диалогового окна конфигурирования для осей и основных параметров в диалоговых окнах конфигурирования для привода и энкодера. Перенос вступает в силу после запуска или перезапуска ТО или после отказа коммуникации с приводом.

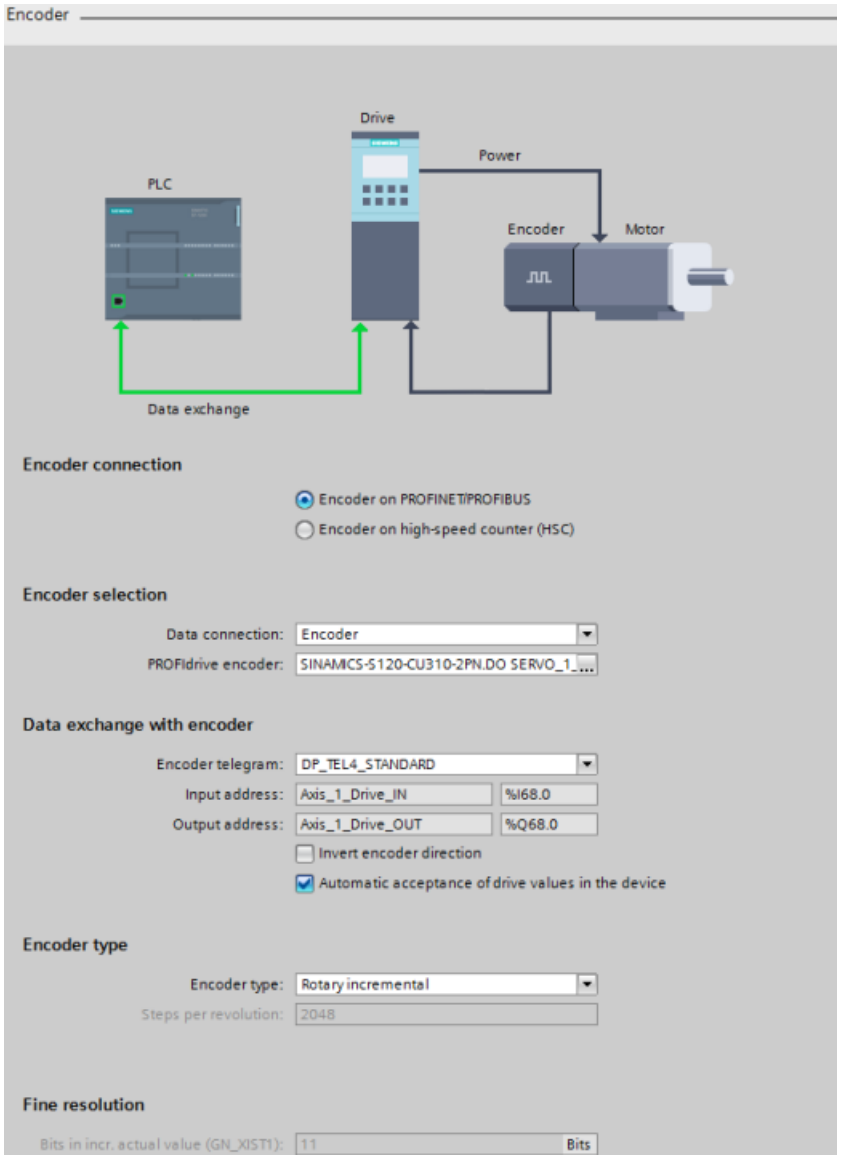
В TIA Portal для выбора "Переноса данных" установить флажок "Автоматический перенос значений привода в устройстве". Эти флажки предлагаются в обоих диалоговых окнах конфигурирования для осей и в основных параметрах диалоговых окон "Привод" и "Энкодер".

На следующем рисунке показано диалоговое окно конфигурирования "Привод":



- Если подключен привод SINAMICS:
  - Флажок установлен по умолчанию.
  - Элементы управления "Опорная скорость" и "Максимальная скорость" под флажком недоступны для выбора.
  - Перенос данных RT для данных привода активирован.
  - Если снять флажок, то значения обоих элементов управления под флажком становятся доступными для редактирования.
- Если привод SINAMICS не подключен или подключен другой привод:
  - Флажок по умолчанию не установлен и серый (выбор недоступен).
  - Элементы управления "Опорная скорость" и "Максимальная скорость" под флажком активны и могут редактироваться.
  - Перенос данных RT для данных привода не активирован.

На следующем рисунке показано диалоговое окно конфигурирования "Энкодер":


Encoder
Drive
Power
PLC
Encoder
Motor
Data exchange
Encoder connection
Encoder on PROFINET/PROFIBUS
Encoder on high-speed counter (HSC)
Encoder selection
Data connection:
Encoder
PROFIdrive encoder:
SINAMICS-S120-CU310-2PN.DO SERVO_1_...
Data exchange with encoder
Encoder telegram:
DP_TEL4_STANDARD
Input address:
Axis_1_Drive_IN
%I68.0
Output address:
Axis_1_Drive_OUT
%Q68.0
Invert encoder direction
Automatic acceptance of drive values in the device
Encoder type
Encoder type:
Rotary incremental
Steps per revolution:
2048
Fine resolution
Bits in incr. actual value (GN_XIST1):
11
Bits

- Если подключен энкодер PROFIdrive:
  - Флажок установлен по умолчанию.
  - Элементы управления "Тип датчика" (только "Шаги на оборот") и "Точное разрешение" под флажком недоступны для выбора.
  - Перенос данных RT для данных энкодера активирован.
  - Если снять флажок, то значения обоих элементов управления под флажком становятся доступными для редактирования.
- Если энкодер PROFIdrive не подключен (модули HSC или TM):
  - Флажок по умолчанию не установлен и серый (выбор недоступен).
  - Элементы управления "Тип датчика" и "Точное разрешение" под флажком активны и могут редактироваться.
  - Перенос данных RT для данных привода не активирован.

Выполнение RT переноса данных

Перенос выполняется при активной RT переноса данных:

- При запуске TO (или запуске PLC) или при создании TO (при загрузке TO-DB)
- После отключения питания для привода или энкодера (или сбоя коммуникации) и перезапуска
- Повторный пуск TO (перенос после каждого повторного пуска)

При активной коммуникации RT переноса данных управление перемещением не принимает связанные с данным приводом коммуникационные команды от программы пользователя и сигнализирует недостаток ресурсов в программе пользователя.

Отказ привода прерывает перенос данных при коммуникации RT переноса данных. Если привод снова отвечает, то перенос данных продолжается.

Если устройство возвращает отрицательное подтверждение, то управление перемещением не перезаписывает конфигурацию TO оси. Если RT переноса данных выбрана, но система не завершает перенос данных успешно, то управление перемещением сигнализирует ошибку, и активация устройства невозможна.

Отображение состояния переноса данных и ошибок в процессе переноса данных

Управление перемещением показывает состояние RT переноса данных и сигнализирует ошибку в параметрах и ErrorID/ErrorInfo:

- Управление перемещением сигнализирует состояние переноса данных на оси: Если перенос данных не выполняется или выполняется с ошибками, то управление перемещением сигнализирует ошибку на оси и показывает ErrorID/ ErrorInfo инструкции MC_Power.
- Общие параметры для переноса данных: Производитель, устройство, версия и дата считываются из параметра P964[7]. Параметр P964 - это параметр PROFIdrive, который доступен как на приводе, так и на энкодере.

- Производительность: Управление перемещением выполняет RT переноса данных после запуска или перезапуска. Если конфигурация после этого не меняется, то имеет смысл отключить перенос данных по причине производительности. Существует два решения:
  - Прикладное сохранение значений после переноса данных в RT и выключение переноса данных.
  - Выгрузка значений после переноса данных, передача этих значений в проект и выключение переноса данных.
- Поведение при переносе данных:
  - При запуске/перезапуске переносятся все сконфигурированные для переноса устройства (привод и энкодер).
  - Если устройство на этом этапе не может быть перенесено, то управление перемещением сигнализирует ошибку и статус изменяется на "ADAPTATION_ERROR".
  - При ошибке энкодера активация оси в режиме управления по скорости возможна, так как это режим обычно не использует энкодеров.

Список ErrorID и ErrorInfo при переносе данных можно найти в "ErrorID и ErrorInfo для инструкций управления перемещением".

Перенос данных привода

Перенос данных привода зависит от конкретного устройства и поддерживается только для приводов SINAMICS. Приводы SINAMICS поддерживают перенос данных и сигнализируют ошибку, если RT переноса данных активирована.

Единицы

Перенос данных привода учитывает сконфигурированные в приводе единицы. Значения единиц зависят от DO-типа, DO-функционального модуля и системы единиц.

Перенос данных привода поддерживает только круговые (не линейные) приводы и единицу "1/мин".

Следующие параметры считываются и анализируются:

- p107 (DO-тип)
- p108 (DO-функциональный модуль)
- p505 (SI-единицы или US-единицы)

Выбранный текущий набор данных привода

Привод SINAMICS поддерживает различные наборы данных для энкодера и привода. Привод SINAMICS принимает текущий на момент переноса набор данных. Поэтому считывается "p51" (текущий набор данных привода):

- Опорная скорость идентична для всех наборов данных: "p2000" (опорная скорость) не зависит от набора данных.
- Максимальная скорость не зависит от набора данных: "p1082".
- Привод SINAMICS не проверяет различные максимальные значения различных наборов данных, если они согласуются относительно их управляющего значения в текущем наборе данных. Например, это происходит после переноса данных в другой набор данных.

Тип двигателя

SINAMICS поддерживает два типа двигателей:

- Линейный двигатель
- Двигатель по умолчанию (круговой двигатель)

С базовым управлением перемещением (BMC) привод SINAMICS поддерживает только круговые двигатели.

Если привод SINAMICS сконфигурирован для линейного двигателя (параметры "r108, бит 12"), то управление перемещением после проверки непротиворечивости или после отмены переноса данных возвращает ErrorID.

Параметры

Управление перемещением поддерживает "DO-Servo" и "DO-Vector". Переносятся следующие параметры привода:

TO-DB параметр привода	Параметры SINAMICS
Actor.DriveParameter.ReferenceSpeed	p2000
Actor.DriveParameter.MaxSpeed	p1082

Три TO-DB параметра привода не переносятся, а проверяются на достоверность:

TO-DB параметр сенсора	SINAMICS/PROFIdrive параметр
Actor.type	r108, bit12
Actor.Interface.AddressIn.RID	p922 или p2079
Actor.Interface.AddressOut.RID	p922 или p2079

Значения каждого параметра перечислены в следующих таблицах:

TO-DB параметр привода	Значение
Actor.type	• 0 = Аналоговый • 1 = PROFIdrive • 2 = PTO Примечание: TIA Portal поддерживает только круговой привод. Поэтому правильным является только значение "1".
Actor.Interface.AddressIn.RID	• 0208_0708 = Телеграмма 1 • 0208_070A = Телеграмма 2 • 0208_070C = Телеграмма 3 • 0208_0720 = Телеграмма 4 • 0208_070E = Телеграмма 81 • 0208_0710 = Телеграмма 83
Actor.Interface.AddressOut.RID	• 0208_0709 = Телеграмма 1 • 0208_070B = Телеграмма 2 • 0208_070D = Телеграмма 3 • 0208_0721 = Телеграмма 4 • 0208_070F = Телеграмма 81 • 0208_0711 = Телеграмма 83

SINAMICS/PROFIdrive параметр	Значение
r108, bit12	• 0 = Линейный привод • 1 = круговой привод Примечание: TIA Portal поддерживает только круговой привод.
p922 или p2079	• 1 = Телеграмма 1 • 2 = Телеграмма 2 • 3 = Телеграмма 3 • 4 = Телеграмма 4 • 81 = Телеграмма 81 • 83 = Телеграмма 83

Проверка максимальной скорости

TIA Portal проверяет достоверность TO-DB параметра "DynamicLimits.MaxVelocity". Но при переносе данных проверка возможна, только если система находится в режиме RUN и запрос подтверждения для TIA Portal не выполняется.

Проверка непротиворечивости

Управление перемещением выполняет при первом запуске или перезапуске ТО-оси проверку непротиворечивости. Управление перемещением также отображает ErrorID, если перенос данных привода активирован. Проверка непротиворечивости распространяется на телеграмму, тип двигателя и максимальную скорость:

- Телеграмма в "p922" или "p2079": При расхождении между ТО и конфигурацией привода управление перемещением сигнализирует ошибку.
- Тип двигателя в "r108, bit12" из DO-Servo: При расхождении между ТО и конфигурацией привода управление перемещением сигнализирует ошибку.
- Максимальная скорость не может быть достигнута со сконфигурированными параметрами оси, и управление перемещением сигнализирует ошибку. (Примечание: Управление перемещением не сигнализирует эту ошибку повторно после подтверждения, если конфигурация оси не будет изменена.)

Особый случай:

- Максимальная скорость (p1082) > 2 x опорная скорость (p2000)
- Управление перемещением выполняет внутреннее уменьшение максимальной скорости до "2x" опорная скорость. Запись в конфигурации оси не ограничивается. Но управление перемещением принимает значение "p1082" и сигнализирует ошибку.

Перенос данных датчика

Поддерживается только перенос данных с активным энкодером (p0979).

Объем переносимых данных

Управление перемещением берет описание фактического значения из "p0979" (включая конфигурацию кругового или линейного энкодера) и проверяет параметр для типа энкодера ("инкрементальный" или "абсолютный") в TIA Portal. Управление перемещением не принимает и не анализирует опорное значение NIST в телеграмме 83.

Сопоставление энкодера в телеграмме с энкодером в PLC и приводе

- Управление перемещением конфигурирует назначение энкодера в PLC с фактическим значением в телеграмме (фактическое значение 1 или фактическое значение 2) в VREF.RID из двух ТО-DB параметров:
  - <axis>.Sensor[i].Interface.AddressIn
  - <axis>.Sensor[i].Interface.AddressOut
- Назначение в приводе осуществляется с настройкой индекса в "p979" (модель датчика в SINAMICS).

Параметры

Управление перемещением переносит следующие параметры:

TO-DB параметр датчика	SINAMICS/PROFIdrive параметр
Sensor[i].System	• P979.[1] или P979.[11] • Бит 0 = 0: Круговой энкодер • Бит 0 = 1: Линейный энкодер
Инкрементный круговой датчик	
Sensor[i].Parameter.StepsPerRevolution	P979.[2] или P979.[12]
Sensor[i].Parameter.FineResolutionXist1	P979.[3] или P979.[13]
Инкрементный линейный датчик	
Sensor[i].Parameter.Resolution	P979.[2] или P979.[12]
Sensor[i].Parameter.FineResolutionXist1	P979.[3] или P979.[13]
Абсолютный круговой датчик	
Sensor[i].Parameter.StepsPerRevolution	P979.[2] или P979.[12]
Sensor[i].Parameter.FineResolutionXist1	P979.[3] или P979.[13]
Sensor[i].Parameter.DeterminableRevolutions	P979.[5] или P979.[15]
Sensor[i].Parameter.FineResolutionXist2	P979.[4] или P979.[14]
Абсолютный линейный датчик	
Sensor[i].Parameter.Resolution	P979.[2] или P979.[12]
Sensor[i].Parameter.FineResolutionXist1	P979.[3] или P979.[13]
Sensor[i].Parameter.FineResolutionXist2	P979.[4] или P979.[14]

Управление перемещением не переносит TO-DB параметр датчика, но проверяет его на непротиворечивость:

TO-DB параметр датчика	SINAMICS/PROFIdrive параметр
Sensor[i].Type	• P979.[5] или P979.[15] • Бит 0 = 0: Инкрементальный энкодер • Бит 0 > 1: Абсолютный энкодер
Sensor[i].Interface.AddressIn Sensor[i].Interface.AddressOut	p922 или p2079

Можно использовать абсолютный энкодер как инкрементальный энкодер, но нельзя использовать инкрементальный энкодер как абсолютный энкодер. Управление перемещением сигнализирует несовместимость типа энкодера в TO и соответствующее фактическое значение в телеграмме PROFIdrive как ошибочное. Примечание: С SINAMICS FW V2.6 у интерфейса PROFIdrive есть нулевая метка датчика.

Проверка непротиворечивости

Управление перемещением выполняет при первом запуске или перезапуске ТО-оси проверку непротиворечивости. Управление перемещением также отображает ErrorID, если перенос данных датчика активирован. Проверка непротиворечивости распространяется на телеграмму и тип датчика:

- Телеграмма в "p922" или "p2079": При расхождении между ТО и конфигурацией датчика управление перемещением сигнализирует ошибку.
- Тип датчика в "P979.[5]" или "P979.[15]": Можно использовать абсолютный энкодер как инкрементальный энкодер, но нельзя использовать инкрементальный энкодер как абсолютный энкодер. Если непротиворечивость отсутствует, то управление перемещением сигнализирует ошибку.

Выгружаемые параметры

В следующем списке перечислены все параметры, которые должна быть выгружены из привода в TIA Portal. Для управления этим переносом данных, необходимо расширить TO-DB. Для этого расширения используются следующие структуры TO-DB:

Привод	Тип	По умолчанию	Изменение	Комментарий
Actor.DataAdaptation	DINT	0	R (с перезапуском)	Активация переноса: • 0: НЕТ • 1: ДА

StatusDrive	Тип	Комментарий
StatusDrive.AdaptationState	DINT	Статус переноса: • 0: NOT_ADAPTED: Перенос данных невозможен. • 1: IN_ADAPTATION: Перенос данных запущен только что. • 2: ADAPTED: Данные перенесены. • 3: NOT_APPLICABLE: Перенос не выбран или недоступен для этого привода • 4: ADAPTATION_ERROR: Ошибка при переносе – Ось не может быть разблокирована. – Управление перемещением сигнализирует ошибку конфигурации.

Датчик	Тип	По умолчанию	Изменение	Комментарий
Sensor.DataAdaptation	DINT	0	R (с перезапуском)	Активация переноса: • 0: НЕТ • 1: ДА

StatusSensor	Тип	Комментарий
StatusSensor.AdaptationState	DINT	Статус переноса: • 0: NOT_ADAPTED: Перенос данных невозможен. • 1: IN_ADAPTATION: Перенос данных запущен только что. • 2: ADAPTED: Данные перенесены. • 3: NOT_APPLICABLE: Перенос не выбран или недоступен для этого привода • 4: ADAPTATION_ERROR: Ошибка при переносе – Ось не может быть разблокирована. – Управление перемещением сигнализирует ошибку конфигурации.

ErrorWord	Тип	Комментарий
...	Bool	
Бит 15: Ошибка при переносе	Bool	ErrorID:
...	Bool	

10.3.4.7 Управление осью с помощью технологического модуля (TM) Pulse

TM Pulse это двухканальный модуль импульсных выходов ET 200SP для использования с вентилями и двигателями. Модуль поддерживает либо два канала 24 В DC с 2 А, либо один канал 24 В DC с 4 А.

В режиме работы "Двигатель постоянного тока" модуля TM Pulse 2x24B двигатель может управляться в обоих направлениях с помощью двухполюсного PWM выхода. Цифровой выход может быть назначен как "Внешний сигнал останова" для двигателя.

TM Pulse может предложить следующие функции для управления перемещением:

- Программируемое поведение на выходе для CPU/Master-Stop
- Обнаружение ошибок / диагностика:
  - Отсутствие и низкое напряжение питания L+
  - Короткое замыкание/перегрузка цифрового выхода
  - Короткое замыкание/пониженное напряжение питания датчика
  - Ошибка по перегреву
  - Ошибка параметрирования
  - Ошибка модуля/прошивки

Пример

Для некоторых приложений важным является быстрый переход к определенной позиции, но не требуется абсолютной точности и соблюдения определенного профиля. Если ET 200SP технологический модуль (TM) Pulse используется для управления двигателем, то может быть выбран стандартный двигатель постоянного тока с сервоуправлением. Этот пример применения рассматривает тип использования ET 200SP TM Pulse: "PWM с двигателем постоянного тока". Справочник *ET 200SP технологический модуль (TM) Pulse 2x24B* содержит дополнительную информацию о ET 200SP технологических модулях Pulse.

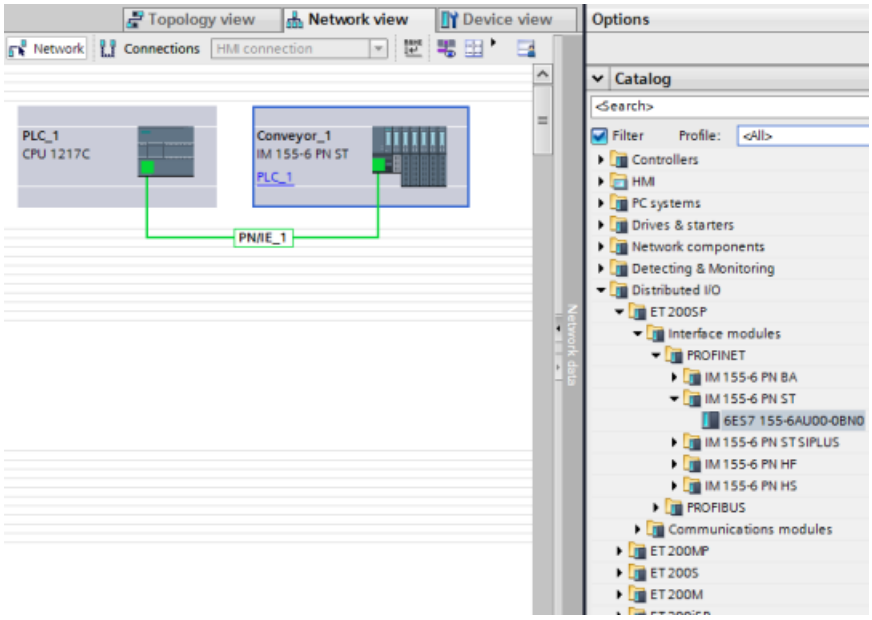
Для полноценного режима регулирования потребуется обратная связь по положению. Необходимо подключить к двигателю энкодер, чтобы отправлять подтверждение в систему управления. В следующем примере показаны три возможности для достижения этого:

- Высокоскоростной счётчик (HSC) в S7-1200
- ET 200SP технологический модуль Count
- ET 200SP технологический модуль PosInput

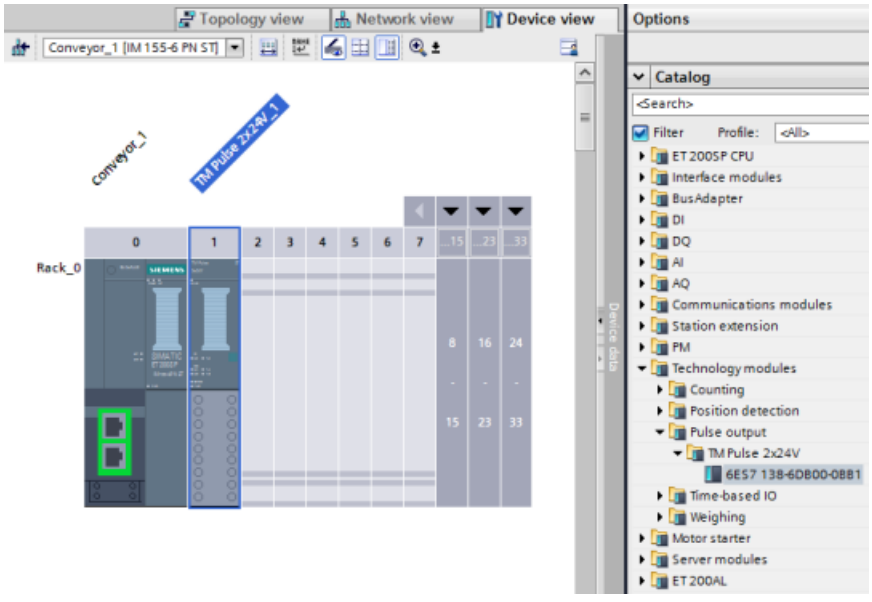
Если требуется перемещение с различными скоростями, то регулирование может функционировать в режиме управления по скорости. В этом режиме обратная связь по положению не нужна.

Для конфигурирования TM Pulse необходимо выполнить следующие действия:

1. Сконфигурировать S7-1200 CPU.
2. Выбрать необходимый ET 200SP интерфейсный модуль и поместить его на вкладку оборудования:



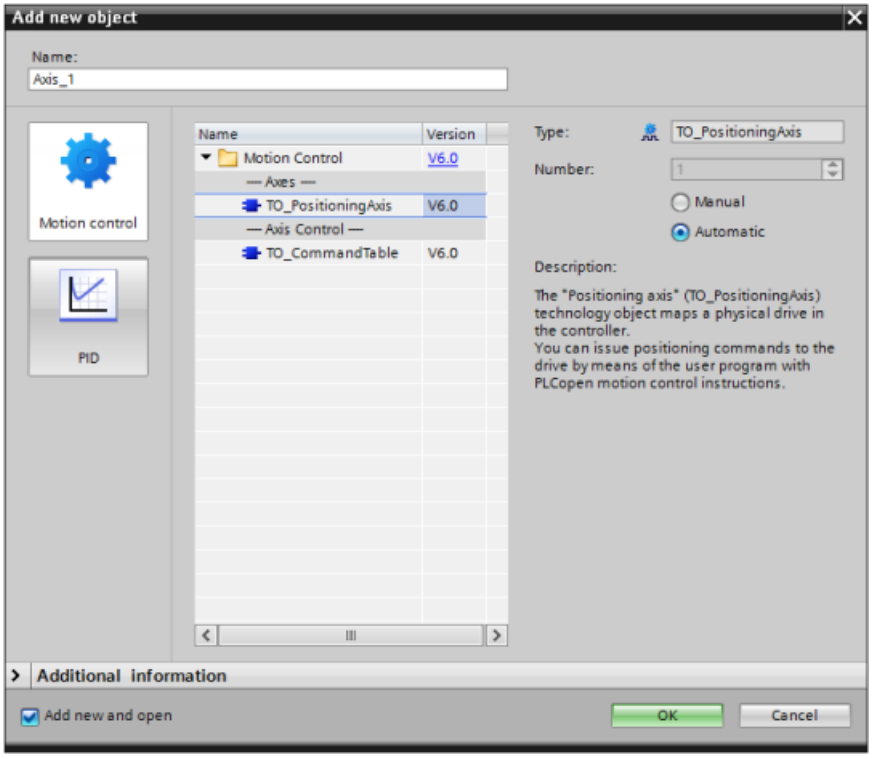
3. Добавить ET 200SP TM Pulse:



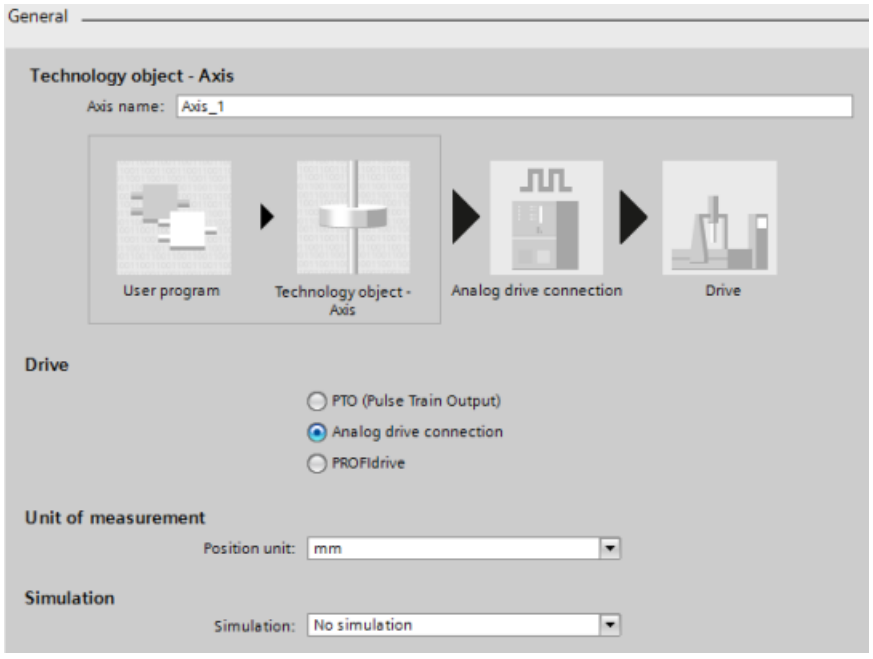
4. В конфигурации каналов выбрать опцию "2 канал (2A)".
5. Установить режим работы "PWM с двигателем постоянного тока".
6. При необходимости настроить параметры диагностики и каналов.

Чтобы настроить ось перемещения с обратной связью по положению, выполнить следующие действия:

1. Если интерфейс конфигурируется с регулируемым управлением перемещением, следует использовать аналоговое управление, не PROFIdrive или PTO. Конфигурирование оси, которая должна регулироваться, выполняется как для оси с аналоговым выходом в качестве входа сервопривода. Вставить на рисунке ниже "TO PositioningAxis":



2. Выбрать в диалоговом окне конфигурирования "Общее" для конфигурации оси вариант "Интерфейс аналогового задания":


General
Technology object - Axis
Axis name: Axis_1
User program
Technology object - Axis
Analog drive connection
Drive
Drive
PTO (Pulse Train Output)
Analog drive connection
PROFIdrive
Unit of measurement
Position unit: mm
Simulation
Simulation: No simulation

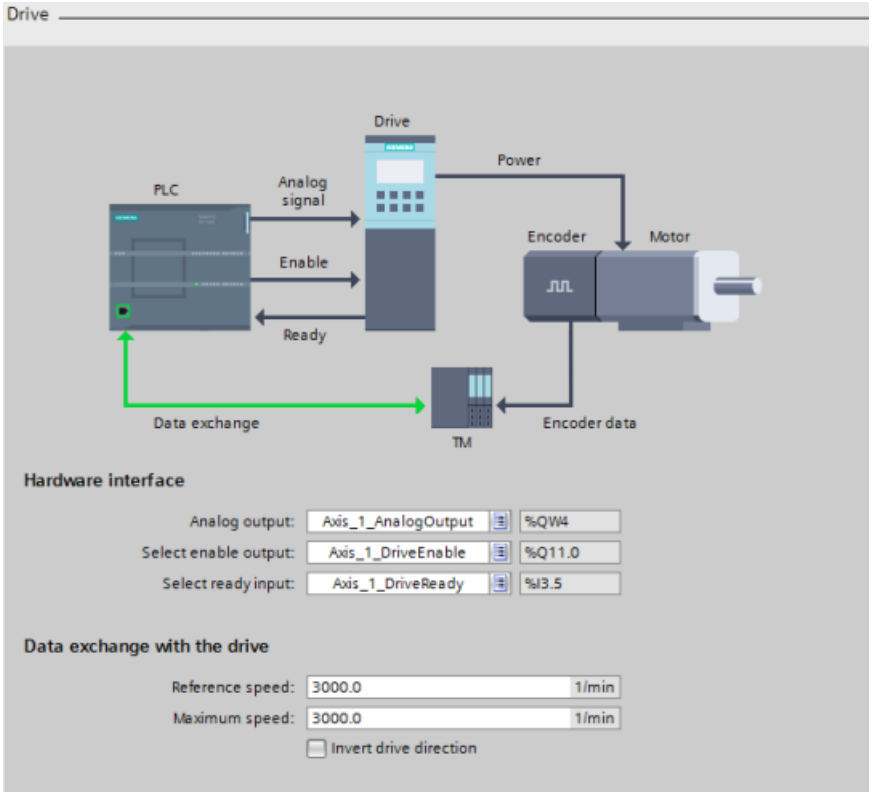
3. Теперь для конфигурации привода потребуются аналоговые выходы и активация привода. Перейти на вкладку оборудования интерфейсного модуля ET 200SP и записи обзора устройства для TM Pulse. С помощью стартового Q адреса для TM Pulse и таблиц интерфейса управления и обратной связи в справочнике *SIMATIC ET 200SP технологический модуль TM Pulse 2x24B (6ES7138-6DB00-0BB1)*, можно определить следующие аналоговые выходы и деблокировки, необходимые для привода:

   – Аналоговый выход

   – Выбрать разрешающий выход

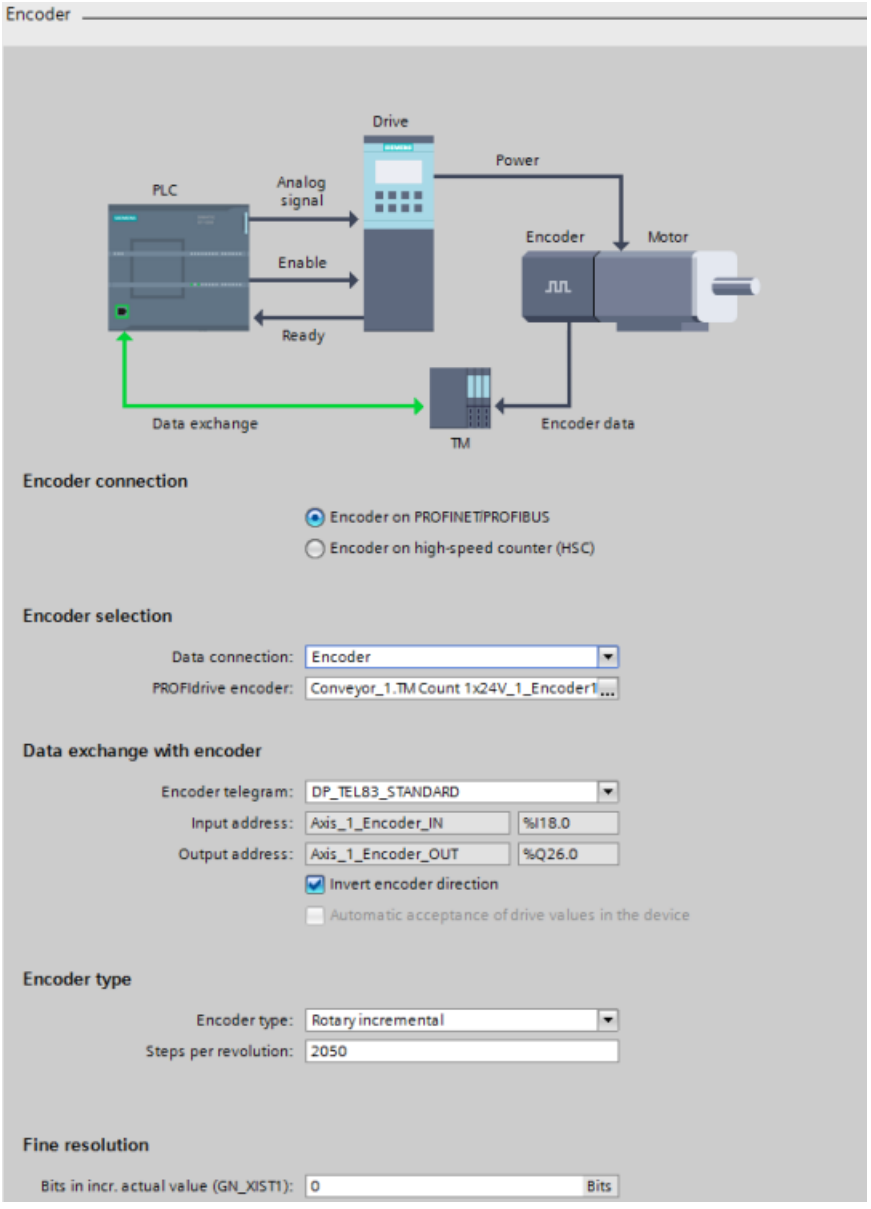
   – Выбрать вход готовности к работе

   В диалоговом окне "Конфигурация привода" для конфигурации оси выбрать I/O аппаратного интерфейса и значения обмена данными:



4. Завершить настройку в диалоговом окне конфигурирования "Энкодер" для конфигурации оси путем выбора одного из следующих энкодеров:

- Технологический модуль Count
- Технологический модуль PosInput
- Высокоскоростной счётчик (HSC)

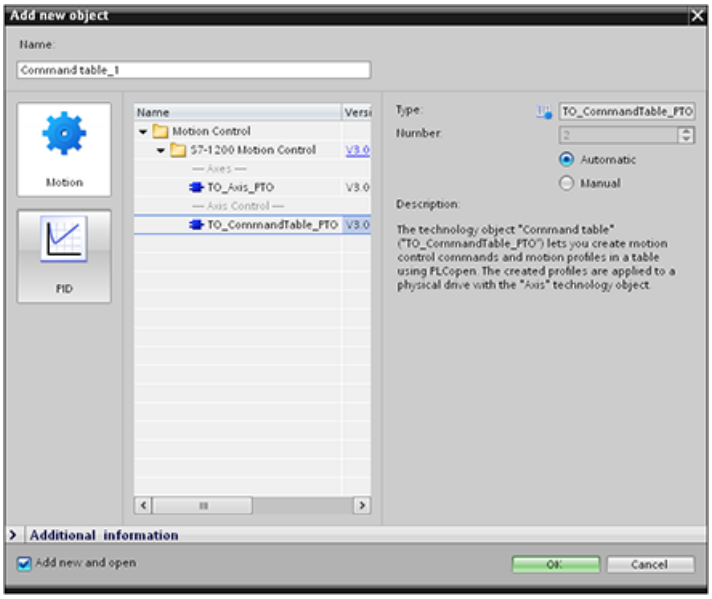


10.3.5 Конфигурирование TO_CommandTable_PTO

С помощью технологических объектов можно сконфигурировать инструкцию MC_CommandTable. В примере ниже показан этот процесс.

Добавление технологического объекта

1. В дереве проекта открыть узел "Технологические объекты" и выбрать "Добавление нового объекта".
2. Выбрать значок "Таблица команд" (переименовать при необходимости), и нажать "OK", чтобы открыть редактор конфигурации для объекта таблицы команд.



Планирование шагов для приложения

Требуемая последовательность движений может быть создана в окне конфигурации "Таблица команд", а для проверки результата можно использовать графическое представление на диаграмме изменений.

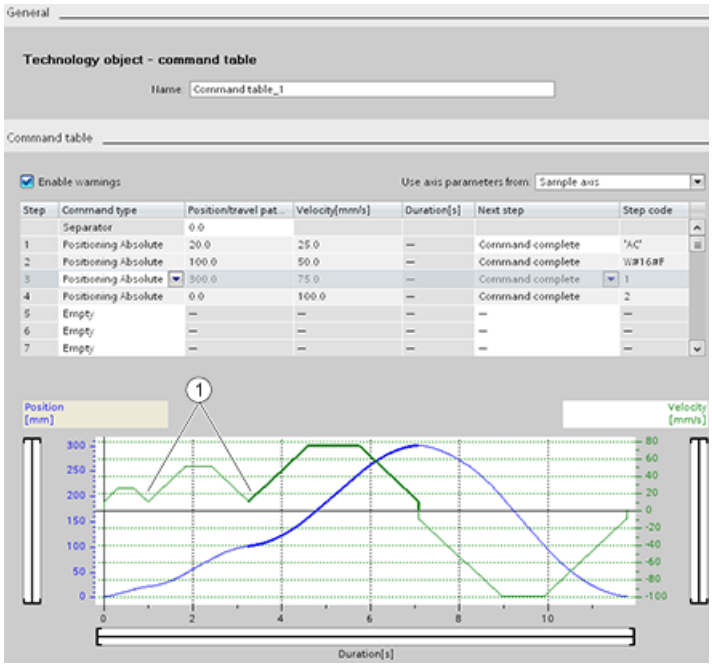
Можно выбирать типы команд, которые должны использоваться для обработки таблицы команд. Может быть введено до 32 шагов. Команды выполняются последовательно, что позволяет без усилий создать сложный профиль движения.

Таблица 10-48 Типы команд MC_CommandTable

Тип команды	Описание
Empty	Эта команда является заполнителем для любой добавляемой команды. Пустая запись игнорируется при обработке таблицы команд.
Halt	Команда останавливает ось. Примечание: Команда выполняется только после команды "Velocity setpoint".
Positioning Relative	Команда учитывает дистанцию при позиционировании оси. Она перемещает ось на указанное расстояние с указанной скоростью.
Positioning Absolute	Команда учитывает положение при позиционировании оси. Она перемещает ось на указанную позицию с заданной скоростью.
Velocity setpoint	Команда перемещает ось с указанной скоростью.

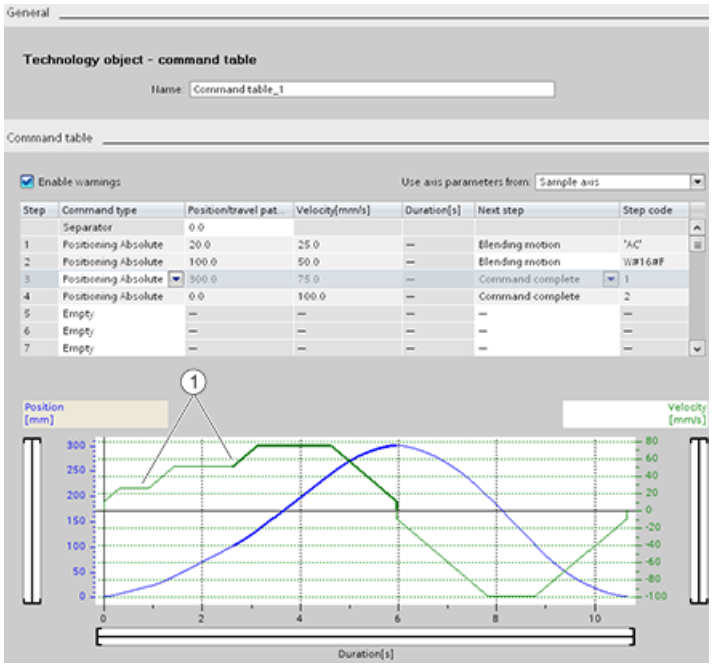
Тип команды	Описание
Wait	Команда ожидает, пока истечет указанный интервал времени. "Wait" не останавливает активное движение перемещения.
Separator	Команда добавляет разделительную линию ("Separator") над выбранной линией. Разделительная линия позволяет определить несколько профилей в одной таблице команд.

На следующем изображении используется "Command complete" в качестве перехода к следующему шагу. Такой тип перехода позволяет устройству выполнить торможение до стартстопной скорости и заново разогнаться в начале следующего шага.



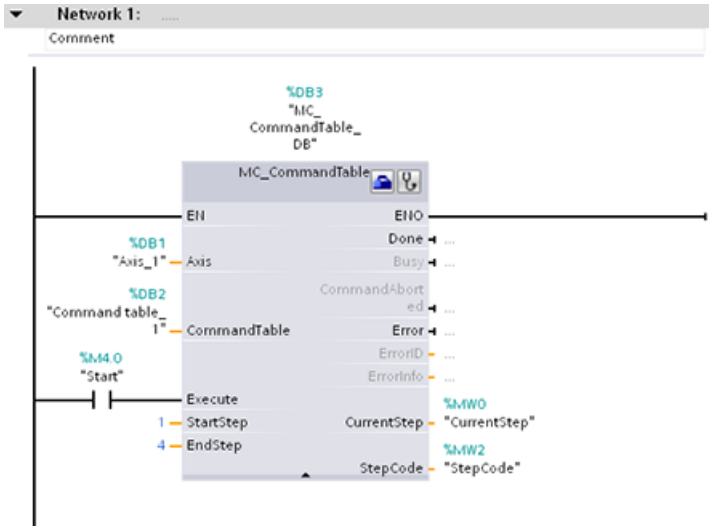
① Ось замедляется между шагами до стартстопной скорости.

На следующем изображении используется "Blending motion" в качестве перехода к следующему шагу. Такой тип перехода позволяет устройству сохранить свою скорость до старта следующего шага, чтобы способствует плавному переходу устройства от одного шага к следующему. Использование "Blending motion" позволяет уменьшить общую продолжительность для полного выполнения профиля. Без плавного перехода выполнение примера занимает семь секунд. С плавным переходом время выполнения сокращается на одну секунду до шести секунд.



① Ось движется непрерывно и разгоняется или замедляется до скорости следующего шага, что позволяет экономить время и уменьшить механический износ.

Принцип работы таблицы команд, как будет показано ниже, управляется инструкцией MC_CommandTable:



10.3.6 Принцип работы управления перемещением для S7-1200

10.3.6.1 Используемые для управления перемещением выходы CPU

CPU предлагает четыре генератора импульсов. У каждого генератора импульсов есть импульсный выход и выход направления для управления приводом шагового двигателя или серводвигателя с импульсным интерфейсом. Импульсный выход обеспечивает привод импульсами, необходимыми для движения двигателя. Выход направления управляет направлением перемещения привода.

Выход PTO генерирует прямоугольный выход переменной частоты. Генерация импульсов управляется информацией о конфигурации и выполнении, задаваемой через аппаратную конфигурацию и/или SFC / SFB.

На основе выбора пользователя, когда CPU находится в рабочем состоянии RUN, либо значения в образце процесса, либо выходы генератора импульсов управляют цифровыми выходами. В режиме STOP генератор PTO не управляет выходами.

В качестве импульсных выходов и выходов направления можно использовать встроенные выходы CPU или выходы сигнальной платы. Выбор между встроенными выходами CPU и выходами сигнальной платы выполняется при конфигурировании генераторов импульсов (PTO/PWM) на вкладке "Свойства". Только PTO (Pulse Train Output) применяется для управления перемещением.

Таблица ниже показывает назначения ввода-вывода по умолчанию; однако, четыре импульсных генератора могут быть сконфигурированы на любой цифровой выход.

Примечание

Импульсные последовательности не могут использоваться другими операциями в программе пользователя.

Если выходы CPU или сигнальной платы конфигурируются как генераторы импульсов (для PWM или инструкций управления перемещением), то соответствующие адреса вывода больше не управляют выходами. Если программа пользователя записывает значение на выход, который используется как генератор импульсов, то CPU не записывает это значение на физический выход.

Примечание

Выходы направления PTO могут быть разрешены для других задач пользователя в программе.

Для каждого PTO должны быть назначены два выхода: один импульсный выход и один выход направления. Можно использовать только импульсный выход, и не использовать выход направления. Это позволит освободить выход направления для других целей в программе пользователя. Выход не может использоваться одновременно и как выхода направления PTO и в программе пользователя.

Таблица 10-49 Назначение адресов по умолчанию для импульсных выходов и выходов направления

Использование выходов для управления перемещением		
	Импульс	Направление
PTO1		
Встроенные I/O	Q0.0	Q0.1
SB-I/O	Q4.0	Q4.1
PTO2		
Встроенные I/O	Q0.2	Q0.3
SB-I/O	Q4.2 [1]	Q4.3 [1]
PTO3		
Встроенные I/O	Q0.4[2]	Q0.5[2]
SB-I/O	Q4.0	Q4.1
PTO4		
Встроенные I/O	Q0.6[3]	Q0.7[3]
SB-I/O	Q4.2	Q4.3

[1] Выходы Q4.2 и Q4.3 доступны только на SB1222 DQ4.

[2] У CPU 1211C нет выходов Q0.4, Q0.5, Q0.6 или Q0.7. Поэтому нельзя использовать эти выходы для CPU 1211C.

[3] У CPU 1212C нет выходов Q0.6 и Q0.7. Поэтому нельзя использовать эти выходы для CPU 1212C.

[4] Эта таблица относится к PTO функциям CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C и CPU 1217C.

Интерфейс привода

Для управления перемещением в качестве опции для интерфейса привода могут быть настроены состояния "Привод разблокирован" и "Привод готов". Если используется интерфейс привода, то цифровой выход для "Привод разблокирован" и цифровой вход для "Привод готов" могут выбираться свободно.

Примечание

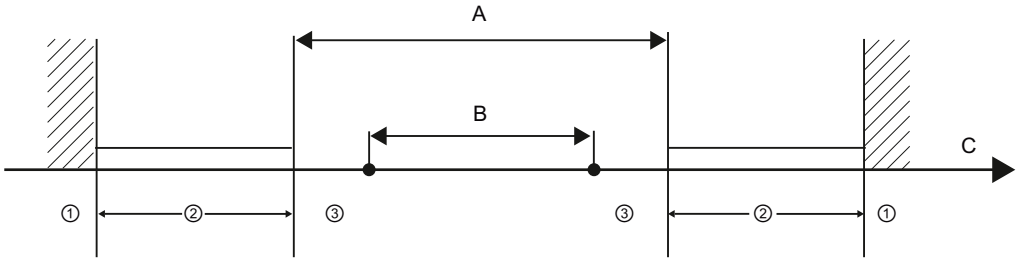
Прошивка берет на себя управление через соответствующие импульсные входы и выходы направления, если PTO (Pulse Train Output) был выбран и назначен оси.

Вместе с этим переходом функции управления, связи между образом процесса и выходом I/O также разрываются. Хотя у пользователя и есть возможность записи в образ процесса импульсных выходов и выходов направления через программу пользователя или таблицу текущего контроля, но эти данные не передаются на выходы. Соответственно невозможен и мониторинг выходов через программу пользователя или таблицу текущего контроля. Считываемая информация просто отражает значение образа процесса и не соответствует фактическому состоянию выходов.

Для всех других выходов CPU, которые не используются постоянно прошивкой CPU, состоянием выходов можно управлять или контролировать, как обычно, через образ процесса.

10.3.6.2 Аппаратные и программные конечные выключатели для управления перемещением

Аппаратные и программные конечные выключатели ограничивают допустимый диапазон перемещения и рабочую область оси.



①	Механический останов	A	Допустимый диапазон перемещения оси
②	Нижнее и верхнее аппаратные предельные значения	B	Рабочая область оси
③	Нижнее и верхнее программные предельные значения	C	Дистанция

Перед использованием в конфигурации или программе пользователя аппаратные и программные конечные выключатели необходимо активировать. Программные конечные выключатели начинают работать только после реферирования оси.

Аппаратные конечные выключатели

Аппаратные конечные выключатели задают максимальный диапазон перемещения оси. Аппаратные конечные выключатели – это физические выключатели, которые подключаются к поддерживающим прерывания входам CPU. Следует использовать только те аппаратные конечные выключатели, которые после наезда остаются включенными. Это состояние можно отключить только после возвращения в допустимые границы перемещения.

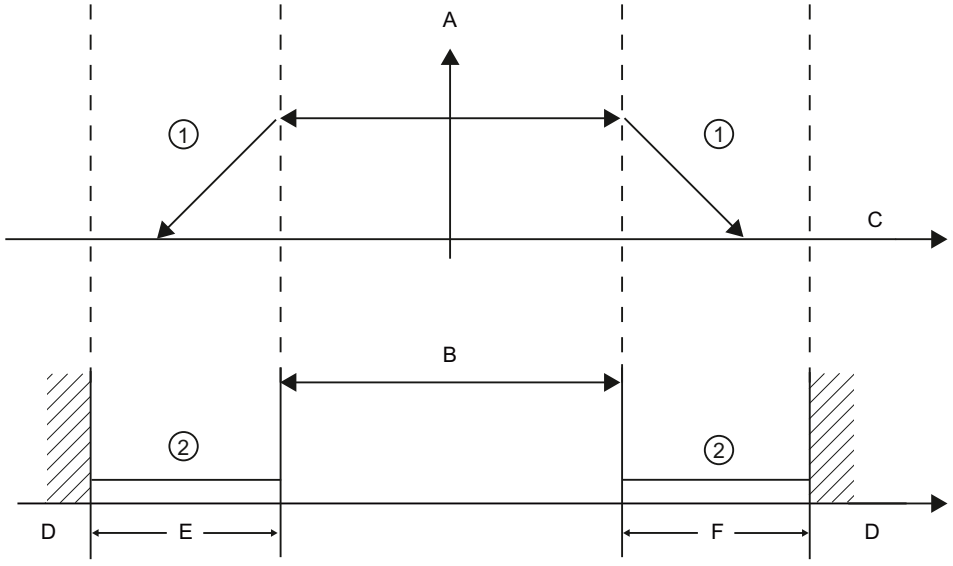
Таблица 10-50 Доступные входы для аппаратных ограничений

Описание	CPU	RPS, LIM- и LIM+ [1]
Встроенные I/O	CPU 1211C	I0.0–I0.5
	CPU 1212C	I0.0–I0.7
	CPU 1214C, CPU 1215C и CPU 1217C	I0.0–!01.3
SB-I/O	Все S7-1200 CPU	I4.0 - I4.3

[1] Вход переключателя референтной точки (Reference Point Switch, RPS), вход отрицательного ограничителя перемещения (Negative Travel Limit, LIM-) и вход положительного ограничителя перемещения (Positive Travel Limit, LIM+)

После наезда на аппаратные конечные выключатели, ось выполняет торможение с настроенным аварийным замедлением до состояния покоя. Указанное аварийное замедление должно быть достаточным для надежного останова оси перед

механическим упором. Поведение оси после наезда на аппаратный конечный выключатель приведено на схеме ниже:



① Ось выполняет торможение с настроенным аварийным замедлением до состояния покоя.

② Диапазон, в котором аппаратные конечные выключатели сигнализируют состояние "наезд".

A [Скорость]

B Допустимый диапазон перемещения

C Дистанция

D Механический останов

E Нижний аппаратный конечный выключатель

F Верхний аппаратный конечный выключатель

⚠ ПРЕДУПРЕЖДЕНИЕ

Риски, связанные с изменениями времени фильтра для каналов цифрового ввода

Если время фильтра для канала цифрового ввода изменяется по сравнению с предыдущей настройкой, то для нового входного значения требуется удержание уровня "0" в течение до 20,0 мс, чтобы фильтр мог полноценно реагировать на новые входы. В течение этого времени короткие импульсы с уровнем "0" продолжительностью менее 20,0 мс не будут обнаружены или подсчитаны.

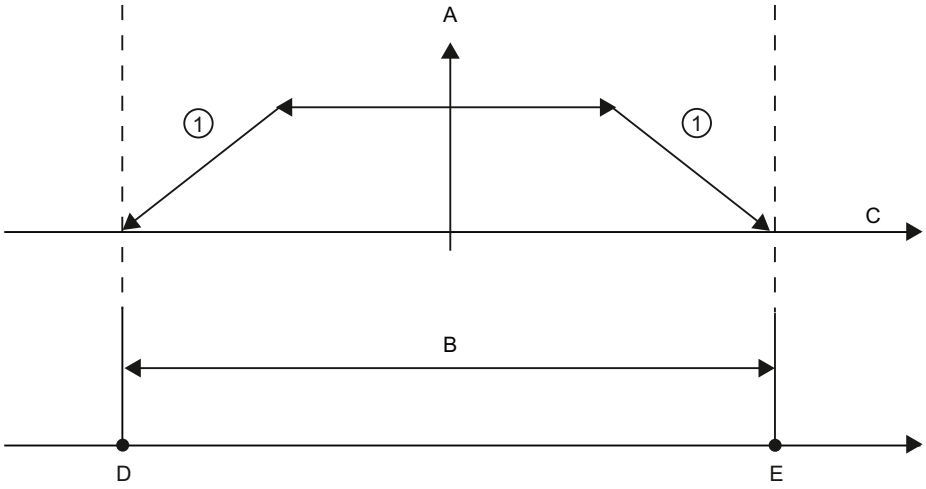
Изменение времени фильтра может вызвать непредсказуемое поведение оборудования или процесса и, как следствие, привести к опасным для жизни телесным повреждениям и/или материальному ущербу.

Чтобы гарантировать, что новое время фильтра сразу вступает в силу, необходимо выключить и снова включить CPU.

Программные конечные выключатели

Программные конечные выключатели ограничивают рабочую область оси. Относительно диапазона перемещения они должны находиться между аппаратными конечными выключателями. Так как позиции программных конечных выключателей легко настраиваются, рабочую область оси можно индивидуально ограничить в зависимости от текущего профиля движения. В отличие от аппаратных конечных выключателей, программные конечные выключатели могут быть реализованы только на программном уровне и для них не требуются собственные переключатели.

После активации программного конечного выключателя, активное движение останавливается на позиции программного конечного выключателя. Оси затормаживается со сконфигурированным замедлением. Поведение оси до достижения программного конечного выключателя показано на схеме ниже:



①	Ось выполняет торможение с настроенным замедлением до состояния покоя.
A	[Скорость]
B	Рабочая область
C	Дистанция
D	Нижний программный конечный выключатель
E	Верхний программный конечный выключатель

Следует использовать дополнительные аппаратные конечные выключатели, если после программных конечных выключателей находится механический упор и существует возможность механического повреждения устройств.

Конфигурация обнаружения фронта при изменении адреса

При конфигурировании предельной позиции или входа переключателя референтной точки для адреса ввода в ТО позиционирующей оси, управление перемещением автоматически определяет прерывания по фронту. Если после предельной позиции или входу переключателя референтной точки присваивается другой адрес, конфигурация с обнаружением фронта для старого адреса остается активной

В диалоговом окне конфигурирования для оси (расширенные параметры, предельные позиции) могут быть добавлены аппаратные и программные конечные выключатели.

Если добавляется вход для аппаратного конечного выключателя, то автоматический активируется обнаружение фронта. Если позже потребуется изменить адрес ввода, то откроется диалоговое окно для обнаружения фронта со следующими опциями:

- Да: Изменяет адрес, активирует обнаружение фронта по новому адресу и выключает обнаружение фронта по старому адресу (по умолчанию)
- Нет: Изменяет адрес, активирует обнаружение фронта по новому адресу и сохраняет обнаружение фронта по старому адресу
- Отмена: Не изменяет адрес и сохраняет текущее состояние обнаружения фронта

Диалоговое окно "Обнаружение фронта"	Изменяет адрес	Активирует обнаружение фронта по новому адресу	Выключает обнаружение фронта по старому адресу
Да (по умолчанию)	Да	Да	Да
Нет	Да	Да	Нет
Отмена	Нет	-/-	Нет

Примечание

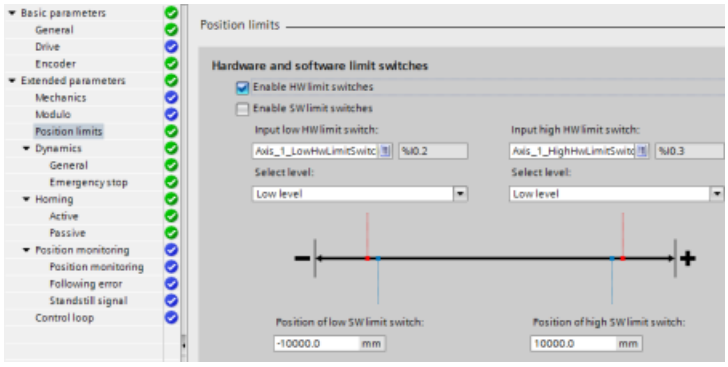
Диалоговое окно не появляется для конфигураций с сервосистемами, которым не требуется обнаружение фронта.

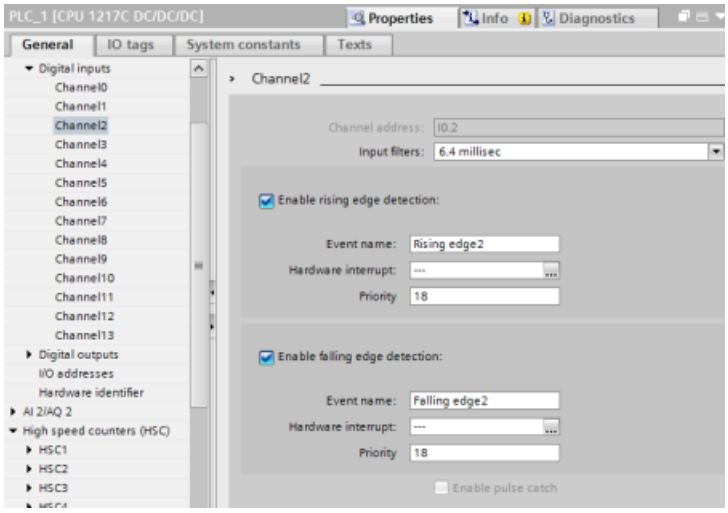
Примечание

Если ОВ связан со входом, то управление перемещением не выполняет описанные выше операции с обнаружением фронта.

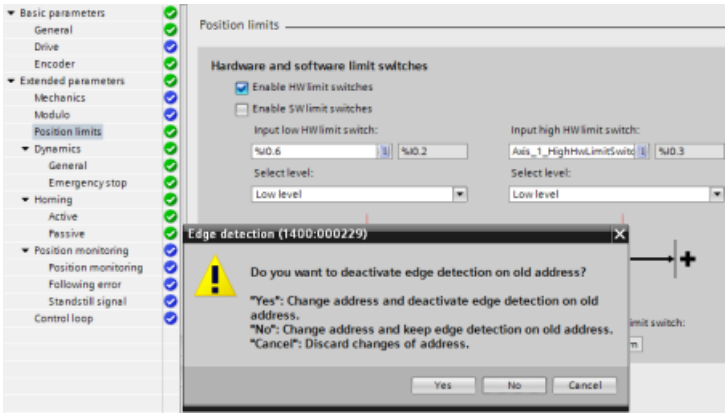
Пример: Указание нового адреса для аппаратного конечного выключателя с обнаружением фронта и выключение обнаружения фронта по старому адресу

1. Текущее состояние: "Вход нижнего аппаратного конечного выключателя" был соединен с I0.2. Эта конфигурация автоматически активирует обнаружение фронта на I0.2. Эта конфигурация появляется и в свойствах CPU, цифровые входы:

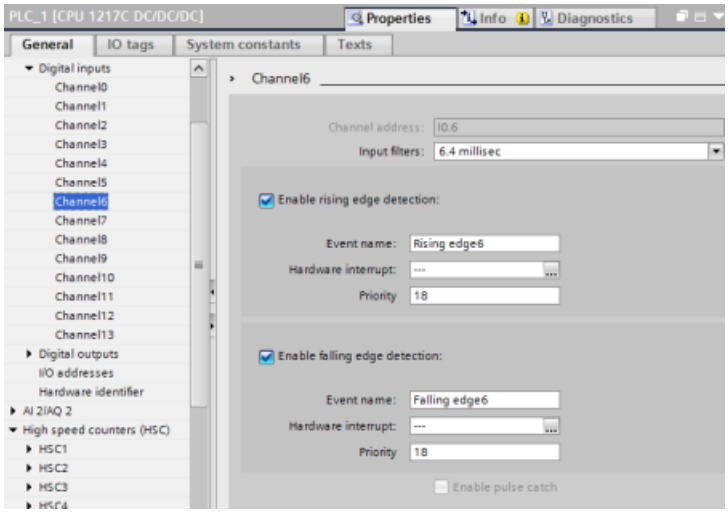


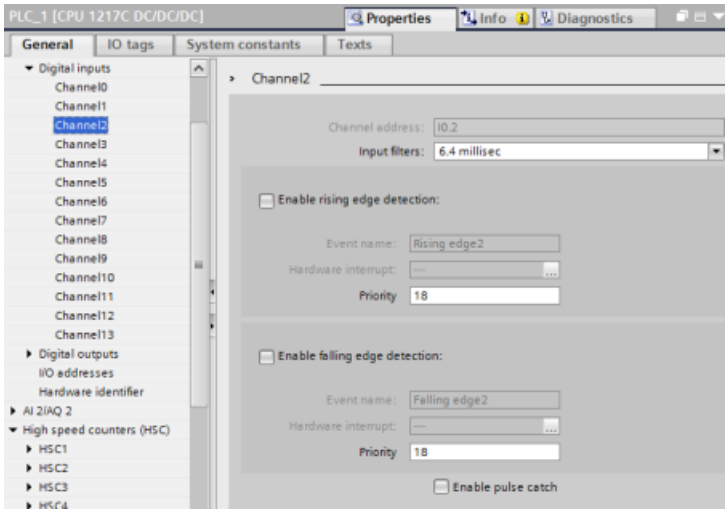


2. Выполняется изменение "Входа нижнего аппаратного конечного выключателя" на I0.6 с подтверждением. Открывается диалоговое окно "Обнаружение фронта". Выполняется выбор: "Да": Изменить адрес и выключить обнаружение фронта по старому адресу.":



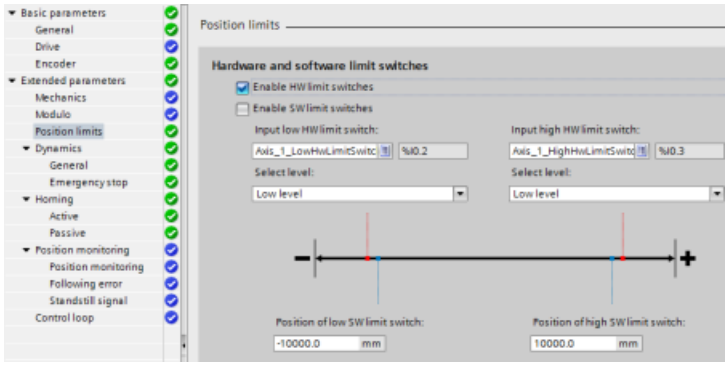
3. CPU применяет новый адрес I0.6 и активирует обнаружение фронта для I0.6. Обнаружение фронта I0.2 в свойствах CPU, цифровые входы, выключается автоматически:

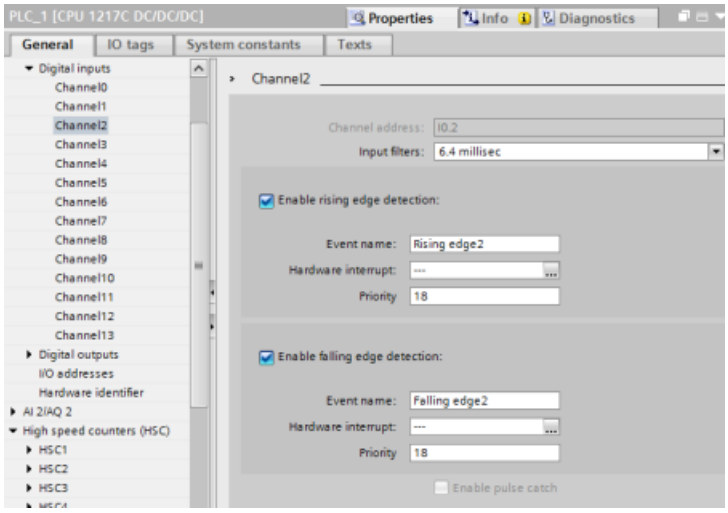




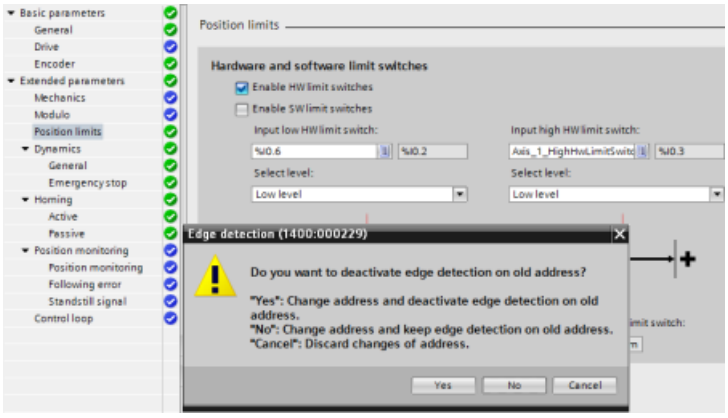
Пример: Указание нового адреса для аппаратного конечного выключателя с обнаружением фронта и сохранение обнаружения фронта по старому адресу

1. Текущее состояние: "Вход нижнего аппаратного конечного выключателя" был соединен с I0.2. Эта конфигурация автоматически активирует обнаружение фронта на I0.2. Эта конфигурация появляется и в свойствах CPU, цифровые входы:

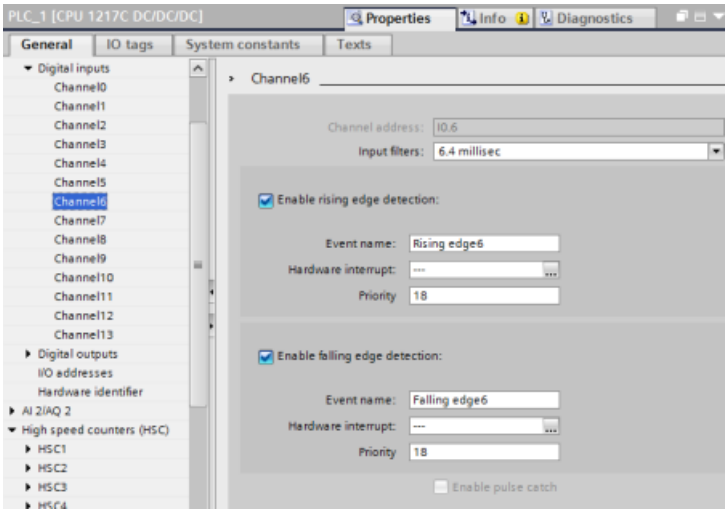


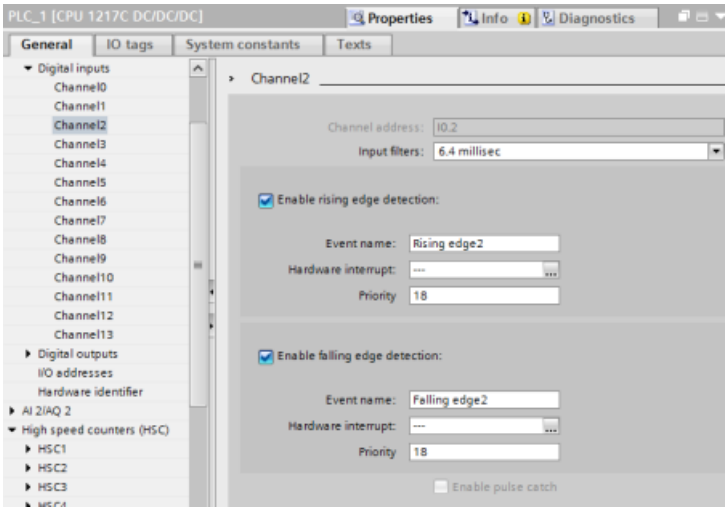


2. Выполняется изменение "Входа нижнего аппаратного конечного выключателя" на I0.6 с подтверждением. Открывается диалоговое окно "Обнаружение фронта". Выполняется выбор: "Нет": Изменить адрес и сохранить обнаружение фронта по старому адресу.":



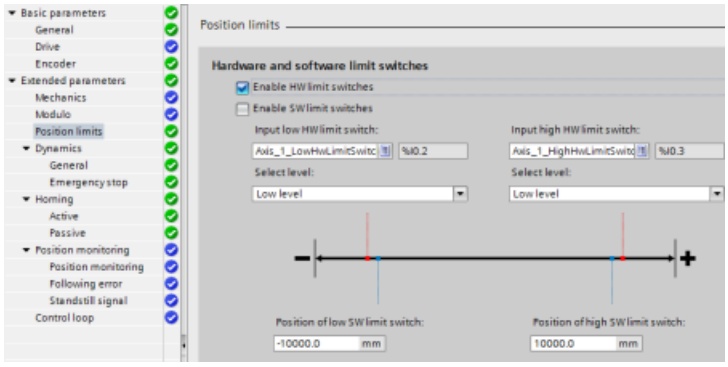
3. CPU применяет новый адрес I0.6 и активирует обнаружение фронта для I0.6. Обнаружение фронта I0.2 в свойствах CPU, цифровые входы, остается включенным:

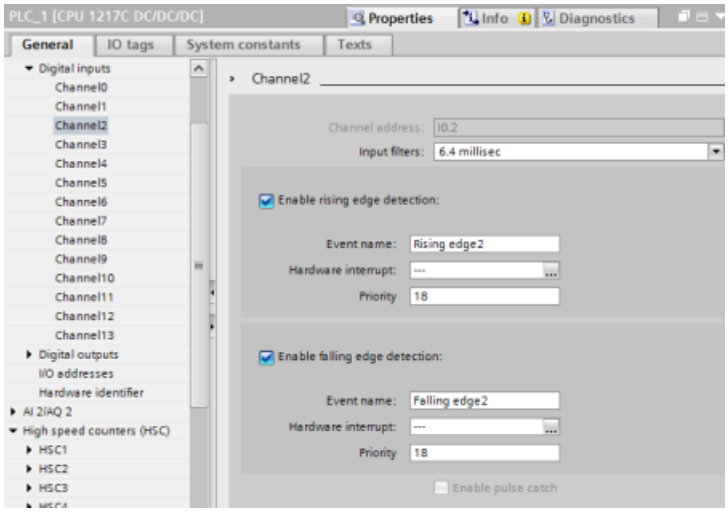




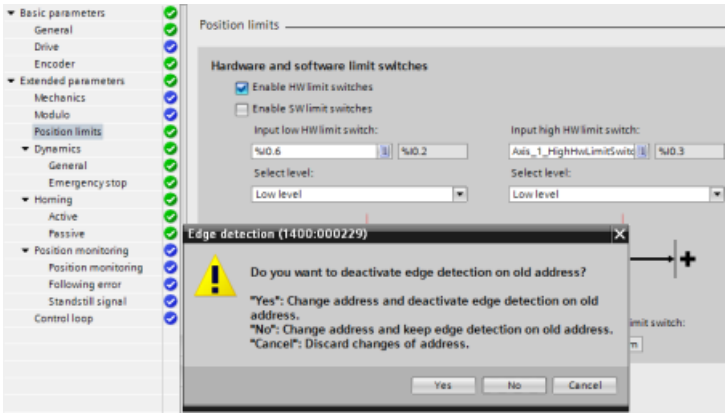
Пример: Отмена изменения на новый адрес для аппаратного конечного выключателя с обнаружением фронта

1. Текущее состояние: "Вход нижнего аппаратного конечного выключателя" был соединен с I0.2. Эта конфигурация автоматически активирует обнаружение фронта на I0.2. Эта конфигурация появляется и в свойствах CPU, цифровые входы:

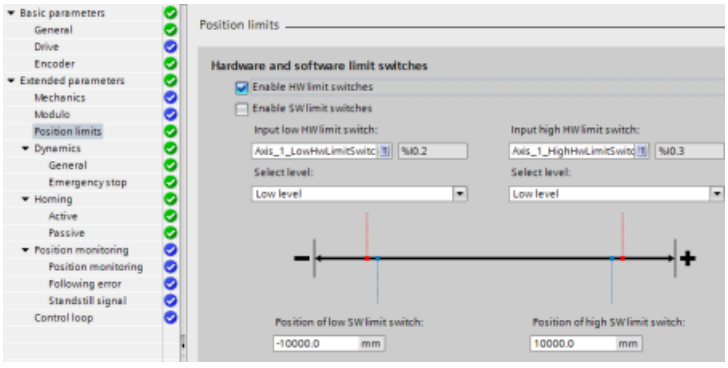


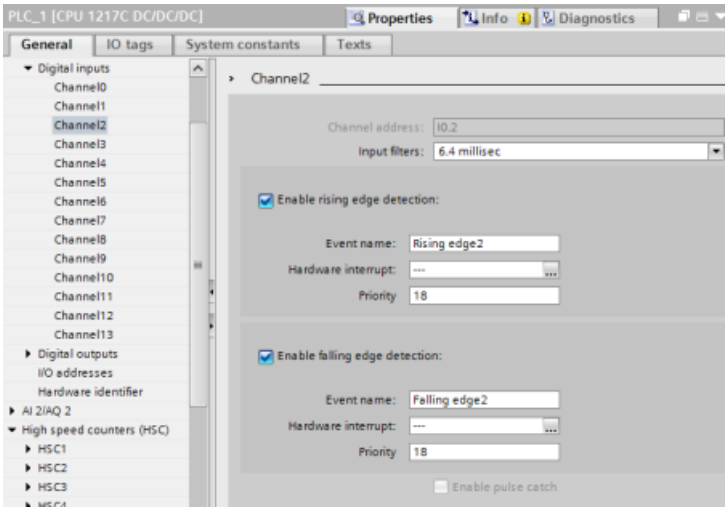


2. Выполняется изменение "Входа нижнего аппаратного конечного выключателя" на I0.6 с подтверждением. Открывается диалоговое окно "Обнаружение фронта". Выполняется выбор: "Отмена":



3. "Вход нижнего аппаратного конечного выключателя" остается сконфигурированным для I0.2 и обнаружение фронта на I0.2 остается включенным:





Дополнительная информация

Программа пользователя может переназначать предельные значения для аппаратных и программных позиций путем включения и выключения аппаратных и программных ограничительных функций. Для выбора используется осевой DB.

- Для включения аппаратной ограничительной функции, вызвать переменную (тег) "Active" (Bool) по следующему адресу DB "<имя оси>/Config/**PositonLimits_HW**". Состояние переменной (тега) "Active" включает или выключает использование аппаратных предельных значений позиции.
- Для включения программной ограничительной функции, вызвать переменную (тег) "Active" (Bool) по следующему адресу DB "<имя оси>/Config/**Position Limits_SW**". Состояние переменной (тега) "Active" включает или выключает использование программных предельных значений позиции.

Для изменения программных предельных значений позиции можно использовать и программу пользователя (например, чтобы сделать отладку станка более гибкой или чтобы сократить время переналадки оборудования). Пользовательская программа может записывать новые значения в переменные (теги) " MinPosition" и " MaxPosition" (физические единицы в формате Real) в DB "<имя оси>/Config/**PositionLimits_SW**".

10.3.6.3 Движение к референтной точке (реферирование)

При реферировании координаты оси согласуются с реальной, физической позицией привода. (Если привод в настоящий момент находится на позиции x , то ось настраивается на позицию x.) У регулируемых по положению осей записи и индикация позиции относятся именно к этим координатам оси.

Примечание

Соответствие между координатами оси и реальной ситуацией является очень важным. Этот шаг необходим, чтобы гарантировать, что абсолютная заданная позиция оси также точно достигается с помощью привода.

Инструкция MC_Home запускает движения оси к референтной точке.

Существует четыре различных функции для реферирования. Первые две функции позволяют пользователю устанавливать текущее положение оси, а вторые две позиционируют ось относительно датчика референтной точки.

- Режим работы 0 - Прямое абсолютное движение к референтной точке: Во время выполнения этот режим сообщает оси точную позицию, где она находится. Внутренняя позиционная переменная устанавливается на значение позиционного входа инструкции для движения к референтной точке. Информация используется при калибровке и настройке оборудования.
  Позиция оси устанавливается независимо от переключателя референтной точки. Активные движения перемещения не прерываются. Значение входного параметра Position инструкции MC_Home сразу же устанавливается как референтная точка оси. Для того, чтобы референтная точка была назначена точной механической позиции, ось на момент установки референтной точки должна находиться на этой позиции в состоянии покоя.
- Режим работы 1 - Прямое абсолютное движение к референтной точке: Это режим работы при выполнении использует внутреннюю позиционную переменную и вставляет значение позиционного входа инструкции для движения к референтной точке в эту переменную. Как правило, это необходимо для учета смещения.
  Позиция оси устанавливается независимо от переключателя референтной точки. Активные движения перемещения не прерываются. Следующее утверждение действительно для позиции оси после движения к референтной точке: Новая позиция оси = текущая позиция оси + значение параметра Position инструкции MC_Home.

- Режим работы 2 - Пассивное движение к референтной точке Если ось при движении переходит через переключатель референтной точки, то текущая позиция устанавливается как референтная точка. Эта функция учитывает нормальный износ машины и боковой зазор шестерен, устраняя потребность в ручной компенсации износа. Позиционный вход инструкции для движения к референтной точке, как и прежде, прибавляется к указанной переключателем референтной точки позиции для простого смещения референтной точки.
  При пассивном реферировании инструкция MC_Home не выполняет движения к референтной точке. Необходимое для этого шага движение перемещения должно быть реализовано через другие инструкции управления перемещением. После обнаружения переключателя референтной точки, ось начинает движение к референтной точке в соответствии с конфигурацией. Активные движения перемещения при запуске пассивного реферирования не прерываются.
- Режим работы 3 - Активное движение к референтной точке: Этот режим работы является точным методом для движения оси к референтной точке. Начальное направление и скорость перемещения настраиваются в расширенных параметрах в конфигурации технологического объекта в реферировании. Это зависит от конфигурации машины. Существует также возможность определить, положительный или отрицательный фронт сигнала переключателя референтной точки соответствует референтной точке. Практически у всех датчиков есть активный диапазон и если положение устойчивого срабатывания используется в качестве сигнала референтной точки, то тогда существует вероятность ошибки референтной точки, так как активный диапазон сигнала срабатывания соответствует диапазону расстояния. При помощи положительного или отрицательного фронта этого сигнала получается намного более точная референтная точка. Как и во всех других режимах работы, значение позиционного входа инструкции для движения к референтной точке прибавляется к аппаратной референтной точке.
  При активном реферировани инструкция MC_Home выполняет необходимое движение к референтной точке. После обнаружения переключателя референтной точки, ось начинает движение к референтной точке в соответствии с конфигурацией. Активные движения перемещения отменяются.

Для режимов работы 0 и 1 вообще не требуется движения оси. Они обычно используются для настройки и калибровки. Для режимов работы 2 и 3 ось должна двигаться и перейти через датчик, сконфигурированный в технологическом объекте "Ось" как переключатель референтной точки. Референтная точка может находиться в рабочей области оси или вне обычной рабочей области оси, но должна лежать в пределах диапазона перемещения.

Конфигурирование параметров для движения к референтной точке

Для конфигурирования параметры активного и пассивного референирования используется окно "Homing". Метод референирования выбирается с помощью входного параметра "Mode" инструкции управления перемещением. Здесь Mode = 2 - это пассивное референирование, а Mode = 3 - активное референирование.

Примечание

Одним из следующих способов убедиться, что при реверсировании машина не совершит наезд на механический упор:

- Выбрать низкую скорость подвода
- Увеличить сконфигурированное значение ускорения / замедления.
- Увеличить расстояние между аппаратным конечным выключателем и механическим упором.

Таблица 10-51 Параметры конфигурации для движения оси к референтной точке

Параметры	Описание
Вход переключателя референтной точки (активное и пассивное референирование)	Из выпадающего списка выбрать цифровой вход для переключателя референтной точки. Вход должен быть рассчитан на поддержку прерываний. Встроенные входы CPU и входы вставленной сигнальной платы могут быть выбраны как входы для переключателя референтной точки. Время фильтра по умолчанию для цифровых входов составляет 6,4 мс. При использовании в качестве переключателя референтной точки это может приводить к нежелательным задержкам и снижению точности. В зависимости от скорости референирования и чувствительности переключателя референтной точки в определенных обстоятельствах возможен сбой при определении референтной точки. Время фильтра можно настроить в разделе цифровых входов конфигурации устройства "Входной фильтр". Указанное время фильтра должно быть меньше, чем длительность входного сигнала на переключателе референтной точки.
Переключатель референтной точки (активное и пассивное референирование)	• Активное референирование: Выбрать, на какой стороне переключателя референтной точки (верхней или нижней) необходимо референировать ось. В зависимости от стартовой позиции оси и конфигурации параметров для движения к референтной точке последовательность операций для подвода к референтной точке может отличаться от показанной на схеме в окне конфигурации. • Пассивное референирование: При пассивном референировании необходимые для этого шага движения перемещения должны быть реализованы через инструкции управления перемещением. Сторона переключателя референтной точки, на которой выполняется движение к референтной точке, зависит от следующих факторов: – Конфигурация "Переключатель референтной точки" – Текущее направление перемещения при пассивном референировании

Параметры	Описание
Автоматический реверс после достижения аппаратных конечных выключателей (только при активном реферировании)	Установить флажок, чтобы использовать аппаратный конечный выключатель в качестве реверсного кулачка для движения к референтной точке. Для реверсирования необходимо сконфигурировать и активировать аппаратные конечные выключатели. Если в процессе активного реферирования происходит наезд на аппаратный конечный выключатель, то ось останавливается с установленным замедлением (не с аварийным замедлением), после чего выполняет реверс. Затем выполняет поиск переключателя референтной точки в обратном направлении. Если функция реверсирования не будет активна, а ось в процессе активного реферирования достигнет аппаратного конечного выключателя, движение к референтной точке будет отменено с ошибкой, а ось выполнит торможение с аварийным замедлением.
Направление подвода (активное и пассивное реферирование)	При выборе направления определяется "Направление подвода" для активного движения к референтной точке при поиске переключателя референтной точки, а также направление реферирования. Направление реферирования определяет направление движения оси для подвода к сконфигурированной стороне переключателя референтной точки для выполнения движения к референтной точке.
Скорость подвода (только при активном реферировании)	В этом поле необходимо указать скорость, с которой будет осуществляется поиск переключателя референтной точки при реферировании. Предельные значения (независимо от выбранной аппаратуры пользователя): Стартстопная скорость ≤ скорость подвода ≤ максимальная скорость
Скорость реферирования (только при активном реферировании)	В этом поле необходимо указать скорость, с которой ось будет выполнять подвод к переключателю референтной точки при реферировании. Предельные значения (независимо от выбранной аппаратуры пользователя): Стартстопная скорость ≤ скорость реферирования ≤ максимальная скорость
Смещение референтной точки (только при активном реферировании)	Если нужная рефрентная точка не совпадает с позицией переключателя референтной точки, то в этом поле может быть указано смещение референтной точки. Если значение отлично от 0, то после реферирования на переключателе референтной точки ось выполняет следующие действия: 1. Переместить ось на скорости реферирования на значение смещения референтной точки. 2. Если позиция референтной точки достигнута, то позиция оси устанавливается на абсолютную референтную точку. Абсолютная референтная точка указывается через параметр "Position" инструкции управление перемещением "MC_Home". Предельные значения (независимо от выбранной пользователем единицы): -1.0e12 ≤ смещение референтной точки ≤ 1.0e12

Таблица 10-52 Влияющие на движение к референтной точке факторы

Влияющие факторы:			Результат:
Конфигурация **Направление подвода**	**Конфигурация** **Переключатель референтной точки**	**Текущее направление перемещения**	**Реферирование на** **переключателе референтной точки**
Положительное	"Нижняя сторона (отрицательная)"	Положительное направление	Внизу
		Отрицательное направление	Вверху

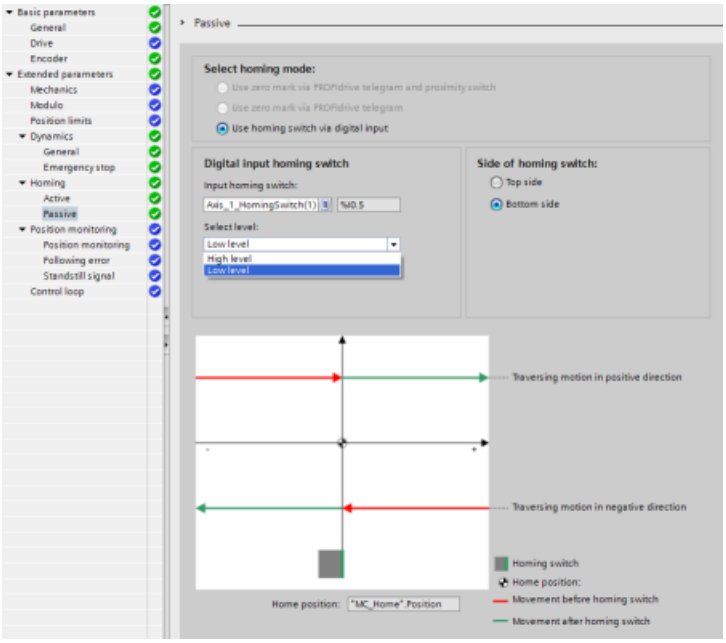
Влияющие факторы:			Результат:
Конфигурация **Направление подвода**	**Конфигурация** **Переключатель референтной точки**	**Текущее направление перемещения**	**Референтирование на переключателе референтной точки**
Положительное	"Верхняя сторона (положительная)"	Положительное направление	Вверху
		Отрицательное направление	Внизу
Отрицательное	"Нижняя сторона (отрицательная)"	Положительное направление	Вверху
		Отрицательное направление	Внизу
Отрицательное	"Верхняя сторона (положительная)"	Положительное направление	Внизу
		Отрицательное направление	Вверху

Уровень переключателя референтной точки для референтирования

В диалоговом окне конфигурирования для осей (референтирование, пассивное или активной) конфигурируется "Цифровой вход переключателе референтной точки". В процессе конфигурирования также можно изменить уровень (высокий или низкий уровень) переключателя референтной точки для регулируемых осей (PROFIdrive и аналоговые оси). Стандартное значение "Высокий уровень".

Пример: Выбор уровня для пассивного переключателя референтной точки

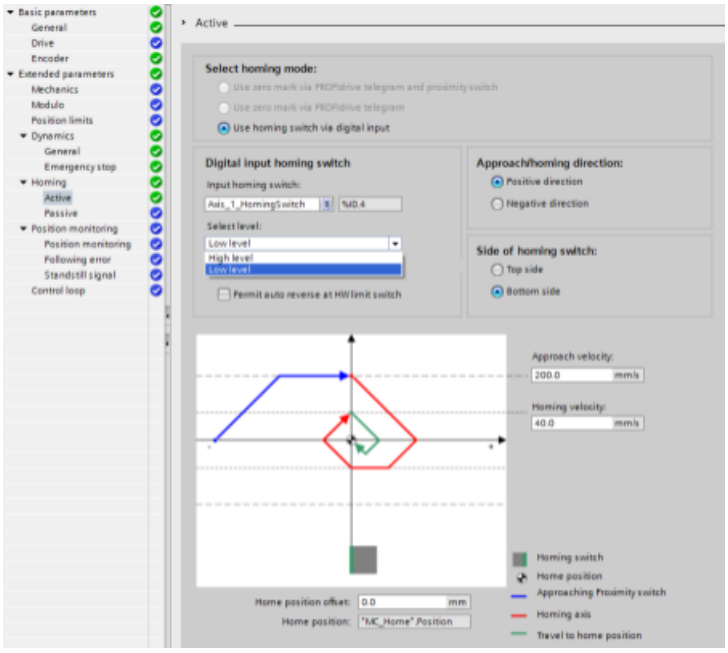
1. Был сконфигурирован проект S7-1200 с аналоговой/PROFIdrive осью и пассивное реферирование. В зависимости от приложения, выбирается "Высокий уровень" или "Низкий уровень" для пассивного переключателя референтной точки:



2. Программа выполняет пассивное реферирование.
3. По завершении пассивного реферирования ось реферирована.

Пример: Выбор уровня для активного переключателя референтной точки

1. Был сконфигурирован проект S7-1200 с аналоговой/PROFIdrive осью и активное реферирование. В зависимости от приложения, выбирается "Высокий уровень" или "Низкий уровень" для активного переключателя референтной точки:



2. Программа выполняет активное реферирование.
3. По завершении активного реферирования ось реферирована.

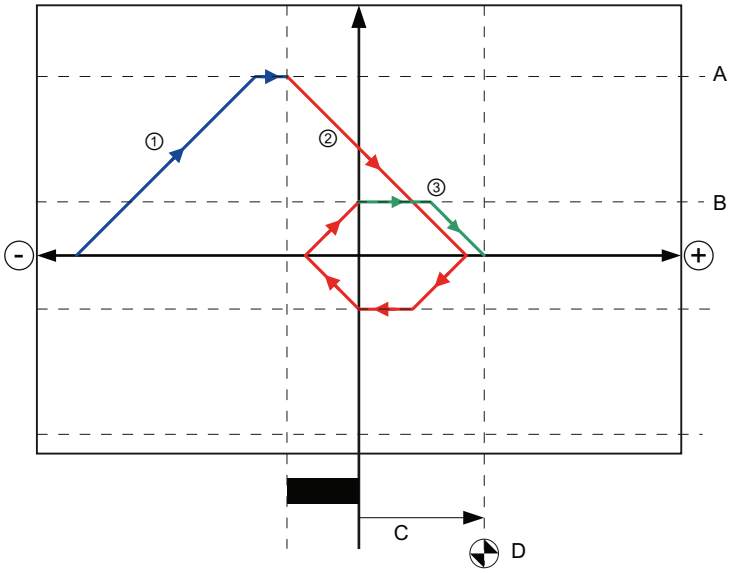
Порядок действий при активном реферировании

Активное реферирование запускается с помощью инструкции управления перемещением "MC_Home" (входной параметр Mode = 3). Входное параметр "Position" здесь указывает абсолютные координаты референтной точки. Альтернативным средством запуска активного реферирования для тестирования является панель управления.

Следующая схема - это пример характеристики для активного реферирования со следующими параметрами конфигурации:

- "Направление подвода" = "Положительное направление подвода"
- "Переключатель референтной точки" = "Верхняя (положительная) сторона"
- Значение "смещения референтной точки" > 0

Таблица 10-53 Эпюра скоростей для движения к референтной точке при управлении перемещением

Принцип работы	Примечания	
	A	Скорость подвода
	B	Скорость реферирования
	C	Координаты референтной точки
	D	Смещение референтной точки

①	Фаза поиска (синий сегмент кривой): После запуска активного реферирования, ось разгоняется до сконфигурированной "Скорости подвода" и выполняет на этой скорости поиск переключателя референтной точки.
②	Подвод к референтной точке (красный сегмент кривой) После обнаружения переключателя референтной точки, ось в данном примере выполняет торможение и реверс, чтобы с установленным значением "Скорости реферирования" продолжить движение к референтной точке на установленной стороне переключателя референтной точки.
③	Перемещение к референтной точке (зеленый сегмент кривой): После реферирования на переключателе референтной точки, ось движется на "Скорости реферирования" к "Координатам референтной точки". После достижения "Координат референтной точки", ось останавливается на позиционном значении, указанном во входном параметре Position инструкция MC_Home.

Примечание

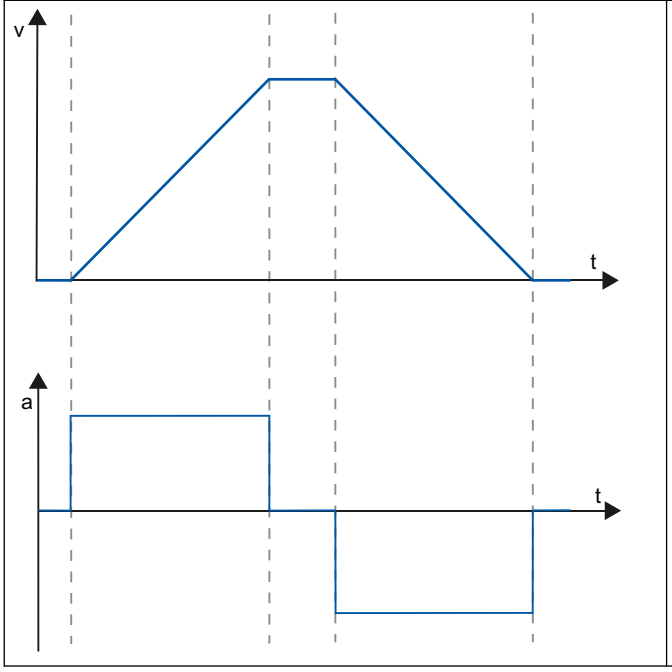
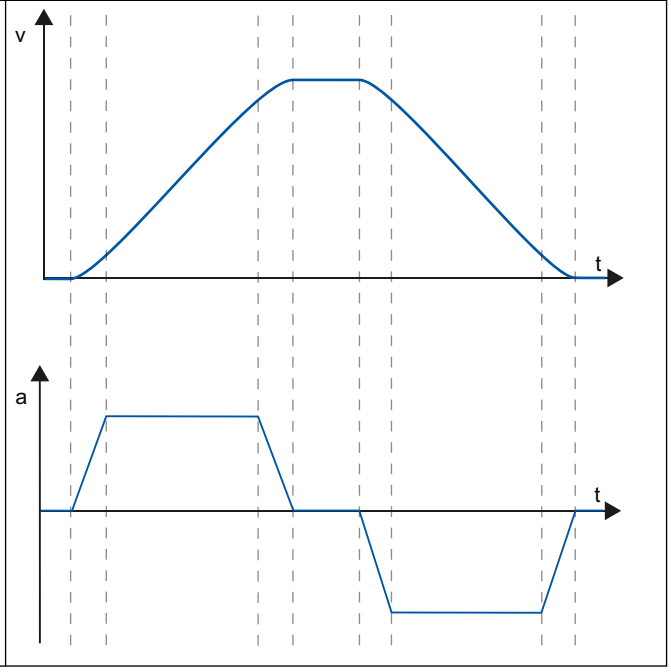
Если поиск референтной точки не работает надлежащим образом, следует проверить входы, назначенные аппаратным предельным значениям или референтной точке. Возможно, что прерывания по фронту для этих входов были отключены в конфигурации устройства.

Проверить данные конфигурации затронутого технологического объекта "Ось", чтобы выяснить, какие входы были назначены для "HW Low Limit Switch Input", "HW High Limit Switch Input" и "Input reference point switch". После открыть конфигурацию устройства CPU и проверить каждый из назначенных входов. Проверить, выбраны ли оба пункта "Активировать обнаружение положительного фронта" и "Активировать обнаружение отрицательного фронта". Если эти свойства не выбраны, то удалить указанные входы в конфигурации оси и повторить выбор.

10.3.6.4 Ограничение рывка

С помощью ограничения рывка можно уменьшить нагрузку на механику во время ускорения и замедления. При активном ограничителе рывка величина ускорения или замедления изменяется не резко, а плавно в рамках переходной фазы. На изображении ниже показано изменение скорости и ускорения с ограничением и без ограничения рывка:

Таблица 10-54 Ограничение рывка

Движение без ограничителя рывка	Движение с ограничителем рывка
	

Ограничение рывка обеспечивает "сглаженный" профиль скорости движения оси. Это гарантирует, например, мягкий запуск и торможение ленточного конвейера.

10.3.7 Инструкции управления перемещением

10.3.7.1 Обзор MC-инструкций

Инструкции управления перемещением используют соответствующий технологический блок данных и назначенный PTO (выход шагового задания) модуля CPU, для управления движением оси.

- Инструкция MC_Power (Страница 737) активирует и деактивирует ось для управления перемещением.
- Инструкция MC_Reset (Страница 740) сбрасывает все ошибки управления перемещением. Все ошибки управления перемещением, которые могут быть квитированы, квитируются.

- Инструкция MC_Home (Страница 741) устанавливает связь между программой управления осью и системой механического позиционирования оси.
- Инструкция MC_Halt (Страница 744) отменяет все процессы перемещения и вызывает останов движения оси. Позиция останова не определена.
- Инструкция MC_MoveAbsolute (Страница 746) запускает движение на абсолютную позицию. Задание завершено после достижения заданной позиции.
- Инструкция MC_MoveRelative (Страница 749) запускает движение позиционирования относительно стартовой позиции.
- Инструкция MC_MoveVelocity (Страница 751) перемещает ось с указанной скоростью.
- Инструкция MC_MoveJog (Страница 754) использует толчковый режим для тестирования и пусковых процессов.
- Инструкция MC_CommandTable (Страница 756) выполняет команды управления осью как последовательность движений.
- Инструкция MC_ChangeDynamic (Страница 759) изменяет динамические настройки оси.
- Инструкция MC_WriteParam (Страница 761) записывает определенные параметры для изменения функциональности оси в программе пользователя.
- Инструкция MC_ReadParam (Страница 763) считывает определенные параметры, указывающие текущую позицию, скорость и т.п. определенной на входе оси.

Версии прошивки CPU

При работе с S7-1200 CPU с прошивкой от версии V4.1, следует выбрать версию V5.0 инструкций перемещения.

При работе с S7-1200 CPU с прошивкой версии V4.0 или ниже, выбрать соответствующую версию V4.0, V3.0, V2.0 или V1.0 инструкций перемещения.

Примечание

Инструкции управления перемещением от V1.0 до V3.0 выполняют активное управление выходом ENO инструкции. При возникновении ошибки в блоке, выход ENO переводится в состояние ВЫКЛ. Ошибка сигнализируется через выходы ERROR, ErrorID и ErrorInfo на блоке. Выход ENO позволяет проанализировать состояние инструкции и последовательно выполнять последующие инструкции.

В инструкциях управления перемещением V4.0 и V5.0 вход EN всегда выводится на выход ENO, независимо от состояния ошибки блока. Это может привести к сбоям в программе, работающей с управлением перемещением V3.0 или более ранних версий и зависящей от состояния ENO. Для решения этой проблемы при использовании управления перемещением V4.0 или выше, следует анализировать состояние инструкции через выходы DONE и ERROR, а не через выход ENO.

Примечание

CPU обсчитывает задачи управления перемещением в "секциях" или сегментах по 10 мс. Пока одна секция выполняется, следующая секция ожидает выполнения в очереди. Если задача управления перемещением для оси прерывается (через выполнение другой новой задачи управления перемещением для оси), новая задача управления перемещением может не выполняться максимум в течение 20 мс (остаток от текущей секции плюс ожидающая в очереди секция).

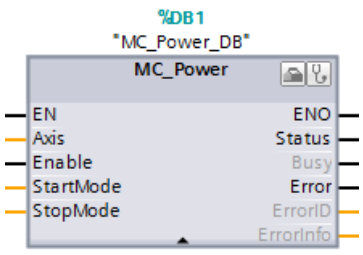
10.3.7.2 Инструкция MC_Power (деблокировка/блокировка оси)

Примечание

Если ось выключается в результате возникновения ошибки, то после устранения и квитирования ошибки она автоматически снова активируется. Условием для этого является сохранение входным параметром Enable во время этой операции значения TRUE.

Таблица 10-55 Инструкция MC_Power

LAD/FBD	SCL	Описание

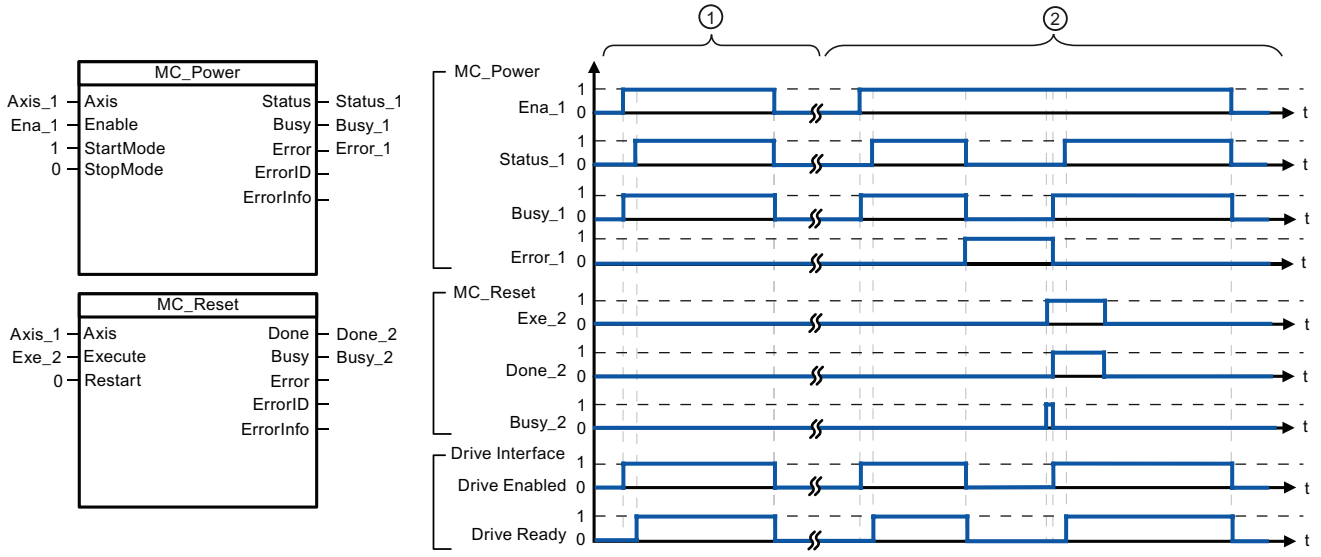
	`"MC_Power_DB"(`
	`Axis:=_multi_fb_in_,`
	`Enable:=_bool_in_,`
	`StartMode:=_int_in_,`
	`StopMode:=_int_in_,`
	`Status=>_bool_out_,`
	`Busy=>_bool_out_,`
	`Error=>_bool_out_,`
	`ErrorID=>_word_out_,`
	
	`ErrorInfo=>_word_out_);`
		Инструкция управления перемещением MC_Power включает или выключает ось. Перед использованием этой возможности, необходимо проверит следующие условия:
		• Технологический объект был сконфигурирован правильно.
		• Препятствующие деблокировке ошибки отсутствуют.
		Выполнение MC_Power не может быть отменено задачей управления перемещением. При выключении оси (входной параметр Enable = FALSE) все задачи управления перемещением для соответствующего технологического объекта отменяются.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "MC_Power_DB" - это имя DB экземпляра.

Таблица 10-56 Параметры для инструкции MC_Power

Параметр и тип данных		Тип данных	Описание
Axis	IN_OUT	TO_Axis	Технологический объект "Ось"
Enable	IN	Bool	• FALSE (по умолчанию): Все текущие задания отменяются в соответствии с установленным "StopMode", и ось останавливается. • TRUE: Управление перемещением пытается включить ось.
StartMode	IN	Int	• 0: С управлением по скорости Примечание: Параметр StartMode обрабатывается только при обнаружении фронта (переход с false на true). • 1: С управлением по положению (по умолчанию)
StopMode	IN	Int	• 0: Аварийное выключение: Если активно требование на выключение оси, то ось выполняет торможение с установленным аварийным замедлением. Ось выключается после достижения состояния покоя. • 1: Немедленный останов: Если активно требование на выключение оси, то ось выключается без замедления. Импульсный выход сразу же останавливается. • 2: Аварийный останов с управлением рывком: Если активно требование на выключение оси, то ось выполняет торможение с установленным замедлением аварийного отключения. Если управление рывком активировано, то учитывается сконфигурированный рывок. Ось выключается после достижения состояния покоя.
Status	OUT	Bool	Состояние деблокировки оси: • FALSE: Ось выключена: – Ось не исполняет заданий управления перемещением и не принимает новых заданий (исключение: задание MC_Reset). – Ось не на референтной точке. – При выключении состояние изменяется на FALSE только после достижения осью состояния покоя. • TRUE: Ось включена: – Ось готова исполнять задания управления перемещением. – При включении оси, состояние изменяется на TRUE только при наличии сигнала "Привод готов". Если интерфейс привода "Привод готов" не был настроен при конфигурации оси, то состояние сразу изменяется на TRUE.
Busy	OUT	Bool	FALSE: Инструкция MC_Power не активна. TRUE: Инструкция MC_Power активна.
Error	OUT	Bool	FALSE: Ошибки отсутствуют TRUE: В инструкции управления перемещением "MC_Power" или в соответствующем технологическом объекте возникла ошибка. Причина ошибки зафиксирована в параметрах ErrorID и ErrorInfo.
ErrorID	OUT	Word	ID ошибки для параметра "Error"
ErrorInfo	OUT	Word	ID информации об ошибке для параметра ErrorID



① Ось включается и после снова выключается. После того, как привод подал сигнал "Привод готов" на CPU, информация об успешном включении отображается при помощи "Status_1".

② После деблокировки оси произошла ошибка, которая привела к выключению оси. Ошибка устраняется и квитируется при помощи "MC_Reset". После ось снова включается.

Для включения оси со сконфигурированным интерфейсом привода, действовать следующим образом:

1. Проверить вышеперечисленные условия.
2. Инициализировать входной параметр "StopMode" с требуемым значением. Установить входной параметр "Enable" на TRUE. Разрешающий выход для "Привод разблокирован" изменяется на TRUE, чтобы включить напряжение для привода. CPU ждет сигнала "Привод готов" от привода. С поступлением сигнала "Привод готов" на сконфигурированный вход готовности CPU, ось деблокируется. Выходной параметр "Status" и переменная (тег) TO <имя оси>.StatusBits.Enable показывают значение TRUE.

Для включения оси без сконфигурированного интерфейса привода, действовать следующим образом:

1. Проверить вышеперечисленные условия.
2. Инициализировать входной параметр "StopMode" с требуемым значением. Установить входной параметр "Enable" на TRUE. Ось включена. Выходной параметр "Status" и переменная (тег) TO <имя оси>.StatusBits.Enable показывают значение TRUE.

Для выключения оси действовать следующим образом:

1. Перевести ось в состояние покоя. На основе переменной (тега) TO <имя оси>.StatusBits.StandStill можно определить, когда ось перейдет в состояние покоя.
2. После достижения состояния покоя установить входной параметр входа "Enable" на FALSE.
3. Если выходной параметр "Busy" и "Status" и переменная (тег) TO <имя оси>.StatusBits.Enable показывают значение FALSE, то выключение оси завершено.

10.3.7.3 Инструкция MC_Reset (подтвердить ошибку)

Таблица 10-57 Инструкция MC_Reset

LAD/FBD	SCL	Описание
	**"MC_Reset_DB"(** **Axis:=_multi_fb_in_,** **Execute:=_bool_in_,** **Restart:=_bool_in_,** **Done=>_bool_out_,** **Busy=>_bool_out_,** **Error=>_bool_out_,** **ErrorID=>_word_out_,** **ErrorInfo=>_word_out_);**	С помощью инструкции MC_Reset выполняется квитирование "Рабочих ошибок с остановкой осей" и "Ошибок конфигурации". Ошибки, которые должны быть квитированы, перечислены в "Списке ErrorIDs и ErrorInfos" в пункте "Метод устранения". Перед использованием инструкции MC_Reset, необходимо устранить причину активной, подлежащей квитированию ошибки конфигурации (напр., путем замены неправильного значения ускорения в ТО "Ось" на правильное значение). От версии 3.0 команда Restart позволяет загрузить конфигурацию осей в рабочую память в рабочем состоянии RUN.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "MC_Reset_DB" - это имя DB экземпляра.

Задача MC_Reset не может быть отменена никакой другой задачей управления перемещением. Новая задача MC_Reset не отменяет никаких активных задач управления перемещением.

Таблица 10-58 Параметры инструкции MC_Reset

Параметр и тип данных		Тип данных	Описание
Axis	IN	TO_Axis_1	Технологический объект "Ось"
Execute	IN	Bool	Запуск задачи по положительному фронту
Restart	IN	Bool	TRUE = Загрузка конфигурацию оси из загружаемой памяти в рабочую память. Команда может быть выполнена только при отключенной оси.
			FALSE = Активные ошибка будут квитированы
Done	OUT	Bool	TRUE = Ошибка была квитирована.
Busy	OUT	Bool	TRUE = Задача выполняется.
Error	OUT	Bool	TRUE = Во время выполнения задачи возникла ошибка. Причина ошибки зафиксирована в параметрах ErrorID и ErrorInfo.
ErrorID	OUTP	Word	ID ошибки для параметра "Error"
ErrorInfo	OUT	Word	ID информации об ошибке для параметра ErrorID

Для квитирования ошибки с MC_Reset необходимо выполнить следующее:

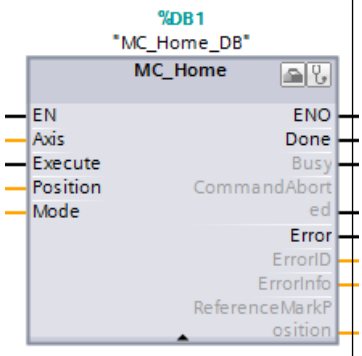
1. Проверить вышеперечисленные условия.
2. Запустить квитирование ошибки по положительному фронту на входном параметре Execute.

3. Ошибка квитирована, если Done = TRUE и переменная (тег) TO <имя оси>.StatusBits.Error = FALSE.
4. Начиная от библиотеки управления перемещением версии 7.0, инструкция MC_Reset может квитировать ошибки в очереди до активации оси программой пользователя. MC_Reset квитирует ошибки в очереди, если коммуникация с приводом и энкодером работает.

10.3.7.4 Инструкция MC_Home (выполнение движения оси к референтной точке)

Таблица 10-59 Инструкция MC_Home

LAD/FBD	SCL	Описание

	`"MC_Home_DB"(`
	`Axis:=_multi_fb_in_,`
	`Execute:=_bool_in_,`
	`Position:=_real_in_,`
	`Mode:=_int_in_,`
	`Done=>_bool_out_,`
	`Busy=>_bool_out_,`
	`CommandAborted=>_bool_out_,`
	`Error=>_bool_out_,`
	`ErrorID=>_word_out_,`
	`ErrorInfo=>_word_out_,`
	`ReferenceMarkPosition=>_real_out_`
	`);`
		С помощью инструкции MC_Home координаты оси согласуются с реальной, физической позицией привода. Для абсолютного позиционирования оси необходимо выполнить движение к референтной точке (референирование):
		Для использования инструкции MC_Home сначала необходимо разблокировать ось.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "MC_Home_DB" - это имя DB экземпляра.

Доступны следующие типы референирования:

- Прямое абсолютное движение к референтной точке (Mode = 0): Текущая позиция оси устанавливается на значение параметра "Position".
- Прямое относительное движение к референтной точке (Mode = 1): Текущая позиция оси смещается на значение параметра "Position".
- Пассивное движение к референтной точке (Mode = 2): При пассивном референировании инструкция MC_Home не выполняет движения к референтной точке. Необходимое для этого шага движение перемещения должно быть реализовано через другие инструкции управления перемещением. После обнаружения переключения референтной точки, ось начинает движение к референтной точке.
- Активное движение к референтной точке (Mode = 3): Движение к референтной точке выполняется автоматически.
- Юстировка абсолютного энкодера (относительная) (Mode = 6): Текущая позиция смещается на значение параметра "MC_Home.Position".
- Юстировка абсолютного энкодера (абсолютная) (Mode = 7): Текущая позиция устанавливается на значение параметра "MC_Home.Position".

Таблица 10-60 Параметры для инструкции MC_Home

Параметр и тип данных		Тип данных	Описание
Axis	IN_OUT	TO_Axis	Технологический объект "Ось"
Execute	IN	Bool	Запуск задачи по положительному фронту
Position	IN	Real	• Mode = 0, 2 и 3 (абсолютная позиция оси после завершения реферирования) • Mode = 1 (поправка для текущей позиции оси) • Mode = 6 (текущая позиция смещается на значение параметра "MC_Home.Position".) • Mode = 7 (текущая позиция устанавливается на значение параметра "MC_Home.Position".) Предельные значения: $-1.0e^{12} \leq$ Position $\leq 1.0e^{12}$
Mode	IN	Int	Тип реферирования: • 0: Прямое абсолютное движение к референтной точке Новая позиция оси - это значение позиции параметра "Position". • 1: Прямое относительное движение к референтной точке Новая позиция оси - это текущая позиция оси + значение позиции параметра "Position". • 2: Пассивное реферирование Движение к референтной точке согласно конфигурации осей. После реферирования значение параметра "Position" устанавливается как новая позиция оси. • 3: Активное реферирование Движение к референтной точке согласно конфигурации осей. После реферирования значение параметра "Position" устанавливается как новая позиция оси. • 6: Текущая позиция смещается на значение параметра "MC_Home.Position". Рассчитанное смещение абсолютного значения остаточно сохраняется в CPU энергонезависимо. (<имя оси>.StatusSensor.AbsEncoderOffset) • 7: Текущая позиция устанавливается на значение параметра "MC_Home.Position". Рассчитанное смещение абсолютного значения остаточно сохраняется в CPU энергонезависимо. (<имя оси>.StatusSensor.AbsEncoderOffset)
Done	OUT	Bool	TRUE = Задача выполнена
Busy	OUT	Bool	TRUE = Задача выполняется.
CommandAborted	OUT	Bool	TRUE = При выполнении задача была отменена другой задачей.
Error	OUT	Bool	TRUE = Во время выполнения задачи возникла ошибка. Причина ошибки зафиксирована в параметрах ErrorID и ErrorInfo.
ErrorID	OUT	Word	ID ошибки для параметра "Error"

Параметр и тип данных		Тип данных	Описание
ErrorInfo	OUT	Word	ID информации об ошибке для параметра ErrorID
ReferenceMarkPosition	OUT	Real	Позиция оси на референтной метке в прежней системе координат.

Выходной параметр "ReferenceMarkPosition": Перед выполнением функции реферирования старая позиция сохраняется и это значение доступно в выходном параметре "ReferenceMarkPosition". Выход демонстрирует следующие значения для различных видов реферирования:

- Активное/пассивное реферирование: Отображается позиция оси на нулевой/референтной метке в прежней системе координат при и после активного/пассивного реферирования.
- Настройка абсолютного энкодера: Отображается прежняя позиция оси при и после настройки абсолютного энкодера.
- Прямое реферирование: Отображается прежняя позиция оси при и после прямого реферирования.

При движении к референтной точке управление перемещением устанавливает позицию оси на новое значение входа "MC_Home.Position". Значения "MC_Home.ReferenceMarkPosition" продолжают действовать, если "MC_Home.Done" = TRUE.

Примечание

Реферирование перестает действовать при следующих условиях

- Блокировка оси при выполнении инструкции MC_Power
- Переключение между автоматическим и ручным режимом
- После запуска активного движения к референтной точке (после успешного выполнения реферирования, движение к референтной точке снова доступно для оси.)
- После выключения и повторного включения CPU
- После перезапуска CPU (RUN-STOP или STOP-RUN)

Для перемещения оси на референтную точку выполнить следующие действия:

1. Проверить вышеперечисленные условия.
2. Инициализировать необходимые входные параметры со значениями и запустить движение к референтной точке с помощью положительного фронта на входном параметре "Execute"
3. Если выходной параметр "Done" и переменная (тег) TO <имя оси>.StatusBits.HomingDone показывают значение TRUE, значит реферирование завершено.

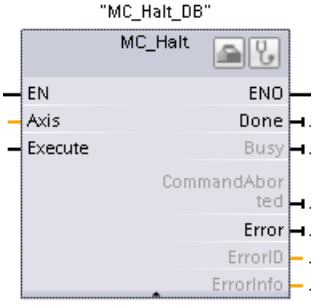
Таблица 10-61 Реакция на внешнее вмешательство

Режим работы	Описание	
0 или 1	Задача MC_Home не может быть отменена никакой другой задачей управления перемещением. Новая задача MC_Home не отменяет никаких активных задач управления перемещением. Позиционные задачи управления перемещением возобновляются после реферирования согласно новой позиции референтной точки (значение во входном параметре Position).	
2	Задача MC_Home может быть отменена следующими задачами управления перемещением: Mode задачи MC_Home = 2, 3: Новая задача MC_Home отменяет следующую активную задачу управления перемещением. Mode задачи MC_Home = 2: Позиционные задачи управления перемещением возобновляются после реферирования согласно новой позиции референтной точки (значение во входном параметре Position).	
3	Задача MC_Home может быть отменена следующими задачами управления перемещением: • MC_Home Mode = 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog	Новая задача MC_Home отменяет следующие активные задачи управления перемещением: • Режим MC_Home = 2, 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog

10.3.7.5 Инструкция MC_Halt (приостановка оси)

Таблица 10-62 Инструкция MC_Halt

LAD/FBD	SCL	Описание

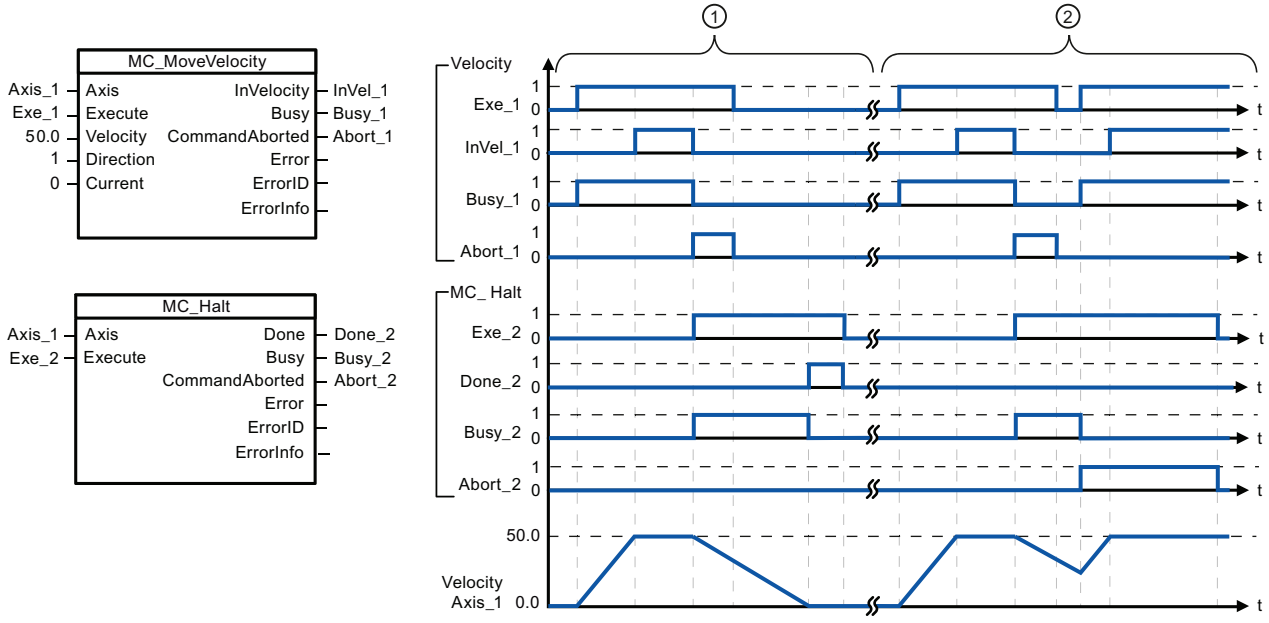
	`"MC_Halt_DB"(`
	`Axis:=_multi_fb_in_,`
	`Execute:=_bool_in_,`
	`Done=>_bool_out_,`
	`Busy=>_bool_out_,`
	`CommandAborted=>_bool_out_,`
	`Error=>_bool_out_,`
	`ErrorID=>_word_out_,`
	`ErrorInfo=>_word_out_);`
		С помощью инструкции MC_Halt останавливаются все движения, и ось переводится в состояние покоя. Позиция состояния покоя не определена.
		Для использования инструкции MC_Halt сначала необходимо разблокировать ось.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "MC_Halt_DB" - это имя DB экземпляра.

Таблица 10-63 Параметры для инструкции MC_Halt

Параметр и тип данных		Тип данных	Описание
Axis	IN	TO_Axis_1	Технологический объект "Ось"
Execute	IN	Bool	Запуск задачи по положительному фронту
Done	OUT	Bool	TRUE = Нулевая скорость достигнута
Busy	OUT	Bool	TRUE = Задача выполняется.
CommandAborted	OUT	Bool	TRUE = При выполнении задача была отменена другой задачей.
Error	OUT	Bool	TRUE = Во время выполнения задачи возникла ошибка. Причина ошибки зафиксирована в параметрах ErrorID и ErrorInfo.
ErrorID	OUT	Word	ID ошибки для параметра "Error"
ErrorInfo	OUT	Word	ID информации об ошибке для параметра ErrorID



Следующие значения были настроены в окне конфигурации "Динамика > Общее": Ускорение = 10,0 и замедление = 5,0

① Ось выполняет торможение до состояния покоя согласно задаче MC_Halt. Состояние покоя оси сигнализируется через "Done_2".

② При торможении оси через задачу MC_Halt, задача отменяется другой задачей управления перемещением. Отмена сигнализируется через "Abort_2".

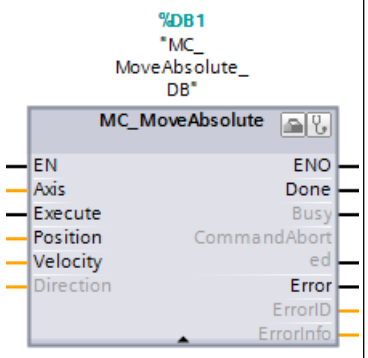
Реакция на внешнее вмешательство

Задача MC_Halt может быть отменена следующими задачами управления перемещением:	Новая задача MC_Halt отменяет следующие активные задачи управления перемещением:
• MC_Home Mode = 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog	• MC_Home Mode = 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog

10.3.7.6 MC_MoveAbsolute (абсолютное позиционирование оси)

Таблица 10-64 Инструкция MC_MoveAbsolute

LAD/FBD	SCL	Описание

	`"MC_MoveAbsolute_DB"(`
	`  Axis:=_multi_fb_in_,`
	`  Execute:=_bool_in_,`
	`  Position:=_real_in_,`
	`  Velocity:=_real_in_,`
	`  Direction:=_int_in_,`
	`  Done=>_bool_out_,`
	`  Busy=>_bool_out_,`
	`  CommandAborted=>_bool_out_,`
	`  Error=>_bool_out_,`
	`  ErrorID=>_word_out_,`
	`  ErrorInfo=>_word_out_);`
		Инструкция MC_MoveAbsolute запускает перемещение оси на абсолютную позицию.
		Для использования инструкции MC_MoveAbsolute сначала необходимо разблокировать и референцировать ось.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "MC_MoveAbsolute_DB" - это имя DB экземпляра.

Таблица 10-65 Параметры для инструкции MC_MoveAbsolute

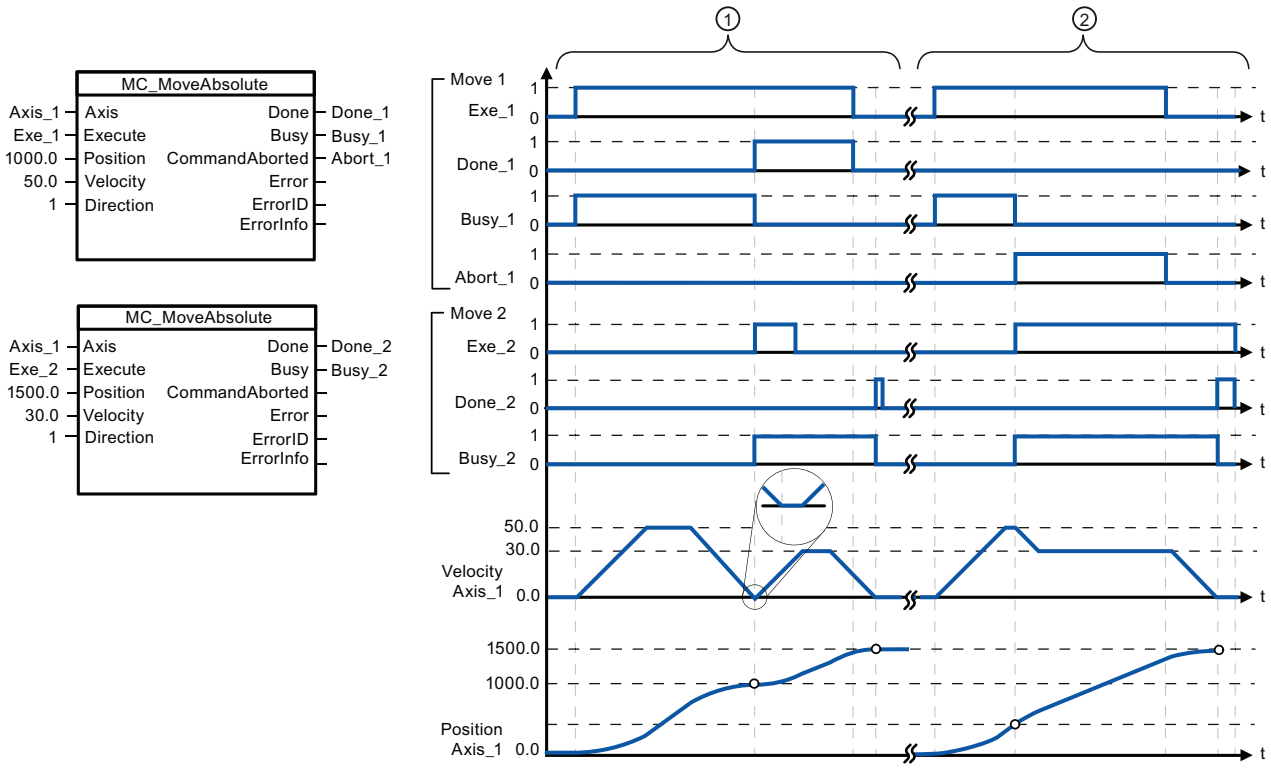
Параметр и тип данных		Тип данных	Описание
Axis	IN	TO_Axis_1	Технологический объект "Ось"
Execute	IN	Bool	Запуск задачи по положительному фронту (по умолчанию: False)
Position	IN	Real	Абсолютная заданная позиция (по умолчанию: 0,0) Предельные значения: $-1.0e^{12} \leq$ Position $\leq 1.0e^{12}$
Velocity	IN	Real	Скорость оси (значение по умолчанию: 10,0) Данная скорость не всегда достигается из-за заданных величин ускорения и замедления и заданной позиции подвода. Предельные значения: Стартстопная скорость ≤ Velocity ≤ Максимальная скорость

Параметр и тип данных		Тип данных	Описание
Direction	IN	Int	Направление вращения (значение по умолчанию: 0)
Done	OUT	Bool	TRUE = Абсолютная заданная позиция достигнута
Busy	OUT	Bool	TRUE = Задача выполняется.
CommandAborted	OUT	Bool	TRUE = При выполнении задача была отменена другой задачей.
Error	OUT	Bool	TRUE = Во время выполнения задачи возникла ошибка. Причина ошибки зафиксирована в параметрах ErrorID и ErrorInfo.
ErrorID	OUT	Word	ID ошибки для параметра "Error" (значение по умолчанию: 0000)
ErrorInfo	OUT	Word	ID информации об ошибке для параметра "ErrorID" (значение по умолчанию: 0000)

Позиционирующая ось может быть сконфигурирована как модульная ось. При использовании модульной оси для выбора направления движения предлагается входной параметр "Direction". Управление перемещением игнорирует вход "Direction", если ось не является модульной осью.

В следующей таблице показаны правильные значения для входа "Direction":

Значение	Перечисления	Комментарий
0	SIGN_OF_VELOCITY	Определяет направление движения
1	POSITIVE	Движение с положительной скоростью
2	NEGATIVE	Движение с отрицательной скоростью
3	SHORTEST_WAY	Движение к цели по кратчайшему пути



Следующие значения были настроены в окне конфигурации "Динамика > Общее": Ускорение = 10,0 и замедление = 10,0

① Задача MC_MoveAbsolute перемещает ось на абсолютную позицию 1000,0. Достижение осью заданной позиции сигнализируется через "Done_1". Если "Done_1" = TRUE, запускается другая задача MC_MoveAbsolute с заданной позицией 1500,0. Из-за времени отклика (напр., время цикла программы пользователя и т.п.) ось на короткое время останавливается (см. увеличенный сегмент). Достижение осью новой заданной позиции сигнализируется через "Done_2".

② Активная задача MC_MoveAbsolute отменяется другой задачей MC_MoveAbsolute. Отмена сигнализируется через "Abort_1". Затем ось с новой скоростью перемещается на новую заданную позицию 1500,0. Достижение осью новой заданной позиции сигнализируется через "Done_2".

Реакция на внешнее вмешательство

Задача MC_MoveAbsolute может быть отменена следующими задачами управления перемещением:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

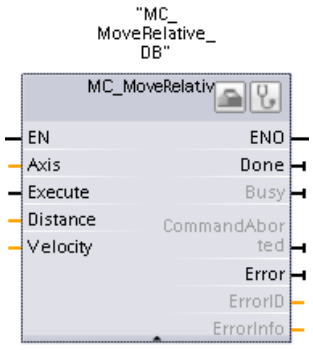
Новая задача MC_MoveAbsolute отменяет следующие активные задачи управления перемещением:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.3.7.7 Инструкция MC_MoveRelative (относительное позиционирование оси)

Таблица 10-66 Инструкция MC_MoveRelative

LAD/FBD	SCL	Описание

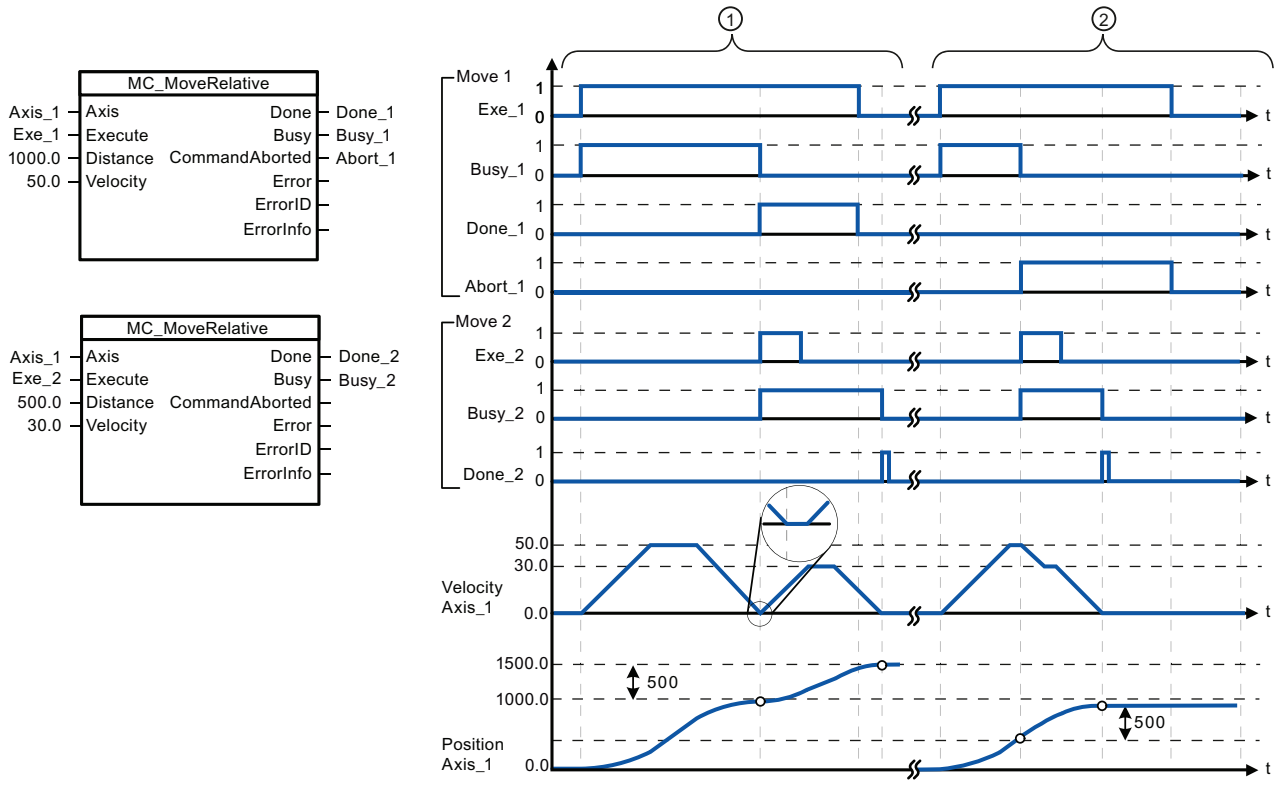
	`"MC_MoveRelative_DB"(`
	`  Axis:=_multi_fb_in_,`
	`  Execute:=_bool_in_,`
	`  Distance:=_real_in_,`
	`  Velocity:=_real_in_,`
	`  Done=>_bool_out_,`
	`  Busy=>_bool_out_,`
	`  CommandAborted=>_bool_out_,`
	`  Error=>_bool_out_,`
	`  ErrorID=>_word_out_,`
	`  ErrorInfo=>_word_out_);`
		Инструкция MC_MoveRelative запускает движение позиционирования относительно стартовой позиции.
		Для использования инструкции MC_MoveRelative сначала необходимо разблокировать ось.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "MC_MoveRelative_DB" - это имя DB экземпляра.

Таблица 10-67 Параметры для инструкции MC_MoveRelative

Параметр и тип данных		Тип данных	Описание
Axis	IN	TO_Axis_1	Технологический объект "Ось"
Execute	IN	Bool	Запуск задачи по положительному фронту (по умолчанию: False)
Distance	IN	Real	Длина перемещения для процесса позиционирования (значение по умолчанию: 0,0) Предельные значения: $-1.0e^{12} \leq$ Distance $\leq 1.0e^{12}$
Velocity	IN	Real	Скорость оси (значение по умолчанию: 10,0) Данная скорость не всегда достигается из-за заданных величин ускорения / замедления и длины перемещения. Предельные значения: Стартстопная скорость ≤ Velocity ≤ Максимальная скорость
Done	OUT	Bool	TRUE = Заданная позиция достигнута
Busy	OUT	Bool	TRUE = Задача выполняется.
CommandAborted	OUT	Bool	TRUE = При выполнении задача была отменена другой задачей.
Error	OUT	Bool	TRUE = Во время выполнения задачи возникла ошибка. Причина ошибки зафиксирована в параметрах ErrorID и ErrorInfo.
ErrorID	OUT	Word	ID ошибки для параметра "Error" (значение по умолчанию: 0000)
ErrorInfo	OUT	Word	ID информации об ошибке для параметра "ErrorID" (значение по умолчанию: 0000)



Следующие значения были настроены в окне конфигурации "Динамика > Общее": Ускорение = 10,0 и замедление = 10,0

① Выполняя задачу MC_MoveRelative, ось проходит расстояние ("Distance") 1000,0. Достижение осью заданной позиции сигнализируется через "Done_1". Если "Done_1" = TRUE, запускается другая задача MC_MoveRelative с длиной перемещения 500,0. Из-за времени отклика (напр., время цикла программы пользователя) ось на короткое время останавливается (см. увеличенный сегмент). Достижение осью новой заданной позиции сигнализируется через "Done_2".

② Активная задача MC_MoveRelative отменяется другой задачей MC_MoveRelative. Отмена сигнализируется через "Abort_1". Затем ось с новой скоростью проходи новое расстояние ("Distance") 500,0. Достижение осью новой заданной позиции сигнализируется через "Done_2".

Реакция на внешнее вмешательство

Задача MC_MoveRelative может быть отменена следующими задачами управления перемещением:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

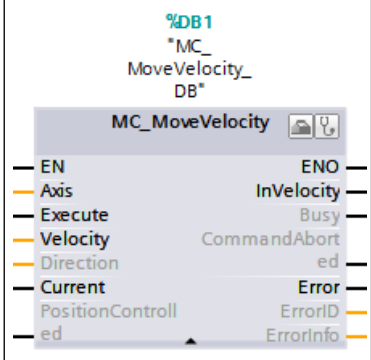
Новая задача MC_MoveRelative отменяет следующие активные задачи управления перемещением:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.3.7.8 Инструкция MC_MoveVelocity (перемещение оси с заданной скоростью)

Таблица 10-68 Инструкция MC_MoveVelocity

LAD/FBD	SCL	Описание

	`"MC_MoveVelocity_DB"(`
	`Axis:=_multi_fb_in_,`
	`Execute:=_bool_in_,`
	`Velocity:=_real_in_,`
	`Direction:=_int_in_,`
	`Current:=_bool_in_,`
	`PositionControlled:=_bool_in_,`
	`InVelocity=>_bool_out_,`
	`Busy=>_bool_out_,`
	`CommandAborted=>_bool_out_,`
	`Error=>_bool_out_,`
	`ErrorID=>_word_out_,`
	`ErrorInfo=>_word_out_);`
		Инструкция MC_MoveVelocity перемещает ось в непрерывном режиме с указанной скоростью.
		Для использования инструкции MC_MoveVelocity сначала необходимо разблокировать ось.

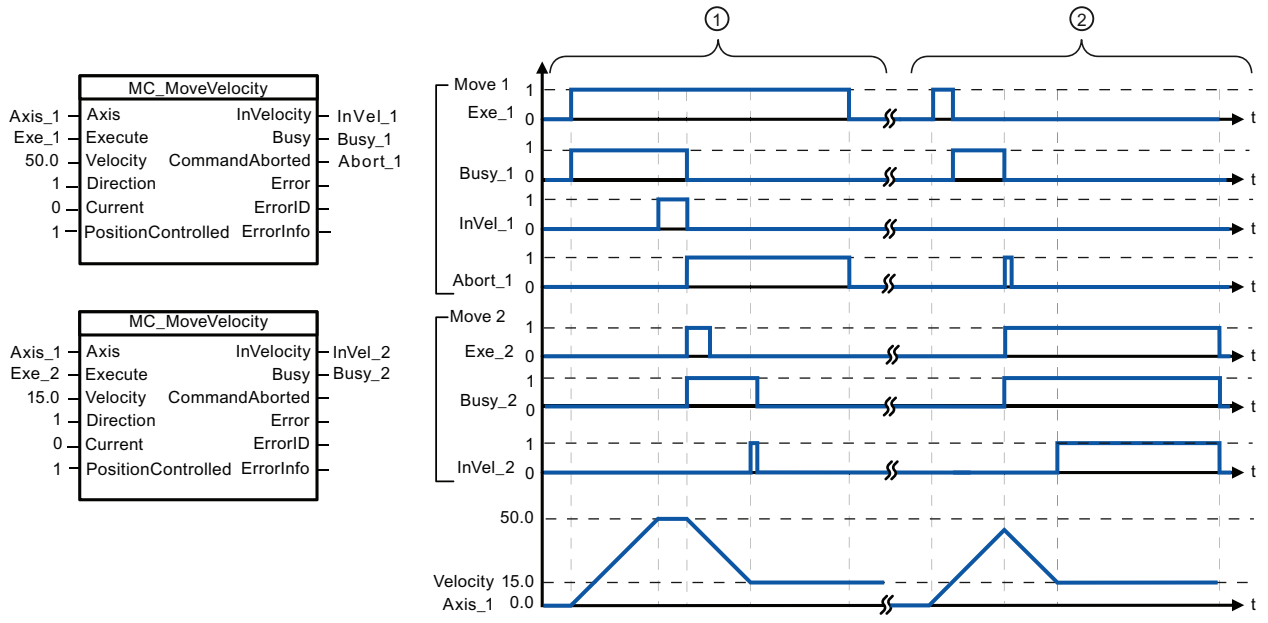
[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "MC_MoveVelocity_DB" - это имя DB экземпляра.

Таблица 10-69 Параметры для инструкции MC_MoveVelocity

Параметр и тип данных		Тип данных	Описание
Axis	IN	TO_SpeedAxis	Технологический объект "Ось"
Execute	IN	Bool	Запуск задачи по положительному фронту (по умолчанию: False)
Velocity	IN	Real	Определение скорости для движения оси (значение по умолчанию: 100.0) Предельные значения: Стартстопная скорость ≤ |Velocity| ≤ Максимальная скорость (Velocity = 0,0 допускается)
Direction	IN	Int	Определение направления: • 0: Направление вращения соответствует знаку значения в параметре "Velocity" (значение по умолчанию) • 1: Положительное направление вращения (знак перед значением параметра "Velocity" игнорируется). • 2: Отрицательное направление вращения (знак перед значением параметра "Velocity" игнорируется).
Current	IN	Bool	Поддержание текущей скорости: • FALSE: "Поддержание текущей скорости" выключено. Используются значения параметров "Velocity" и "Direction". (значение по умолчанию) • TRUE: "Поддержание текущей скорости" включено. Значения параметров "Velocity" и "Direction" не учитываются. Если ось возобновляет движение с текущей скоростью, то параметр "InVelocity" выводит значение TRUE..

Параметр и тип данных		Тип данных	Описание
PositionControlled	IN	Bool	• 0: Управления по скорости • 1: Управления по скорости (значение по умолчанию: true)
InVelocity	OUT	Bool	TRUE: • Если "Current" = FALSE: Была достигнута скорость, заданная в параметре "Velocity". • Если "Current" = TRUE: Ось перемещается с текущей во время запуска скоростью.
Busy	OUT	Bool	TRUE = Задача выполняется.
CommandAborted	OUT	Bool	TRUE = При выполнении задача была отменена другой задачей.
Error	OUT	Bool	TRUE = Во время выполнения задачи возникла ошибка. Причина ошибки зафиксирована в параметрах ErrorID и ErrorInfo.
ErrorID	OUT	Word	ID ошибки для параметра "Error" (значение по умолчанию: 0000)
ErrorInfo	OUT	Word	ID информации об ошибке для параметра "ErrorID" (значение по умолчанию: 0000)



Следующие значения были настроены в окне конфигурации "Динамика > Общее": Ускорение = 10,0 и замедление = 10,0

① Активная задача MC_MoveVelocity сигнализирует через "InVel_1", что заданная скорость была достигнута. После она отменяется другой задачей MC_MoveVelocity. Отмена сигнализируется через "Abort_1". Достижение осью новой заданной скорости 15,0 сигнализируется через "InVel_2". После ось продолжает движение с новой постоянной скоростью.

② Активная задача MC_MoveVelocity отменяется другой задачей MC_MoveVelocity, до достижения своей заданной скорости. Отмена сигнализируется через "Abort_1". Достижение осью новой заданной скорости 15,0 сигнализируется через "InVel_2". После ось продолжает движение с новой постоянной скоростью.

Реакция на внешнее вмешательство

Задача MC_MoveVelocity может быть отменена следующими задачами управления перемещением:	Новая задача MC_MoveVelocity отменяет следующие активные задачи управления перемещением:
• MC_Home Mode = 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog	• MC_Home Mode = 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog

Примечание

Поведение при установленной на ноль скорости (Velocity = 0,0)

Задача MC_MoveVelocity с "Velocity" = 0,0 прерывает (как и задача MC_Halt) активные задачи управления перемещение и останавливает ось с заданным замедлением. После перехода оси в состояние покоя, выходной параметр "InVelocity" как минимум в течение одного программного цикл остается TRUE.

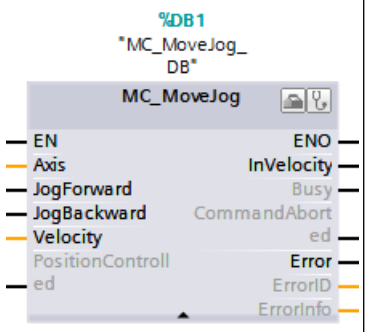
"Busy" выводит в процессе замедления значение TRUE и переходит вместе с "InVelocity" на FALSE. Если параметр "Execute" = TRUE установлен, то "InVelocity" и "Busy" установлены как сохраненные.

С запуском задачи MC_MoveVelocity в технологическом объекте устанавливается бит состояния "SpeedCommand". Бит состояния "ConstantVelocity" устанавливается в состоянии покоя оси. Оба бита подстраиваются под новую ситуацию при запуске нового задания управления перемещением.

10.3.7.9 Инструкция MC_MoveJog (перемещение оси в толчковом режиме)

Таблица 10-70 Инструкция MC_MoveJog

LAD/FBD	SCL	Описание

	`"MC_MoveJog_DB"(`
	`  Axis:=_multi_fb_in_,`
	`  JogForward:=_bool_in_,`
	`  JogBackward:=_bool_in_,`
	`  Velocity:=_real_in_,`
	
	`PositionControlled:=_bool_in_,`
	`  InVelocity=>_bool_out_,`
	`  Busy=>_bool_out_,`
	`  CommandAborted=>_bool_out_,`
	`  Error=>_bool_out_,`
	`  ErrorID=>_word_out_,`
	`  ErrorInfo=>_word_out_);`
		Инструкция MC_MoveJog непрерывно перемещает ось в толчковом режиме с указанной скоростью. Эта инструкция обычно используется при тестировании и вводе в эксплуатацию.
		Для использования инструкции MC_MoveJog сначала необходимо разблокировать ось.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

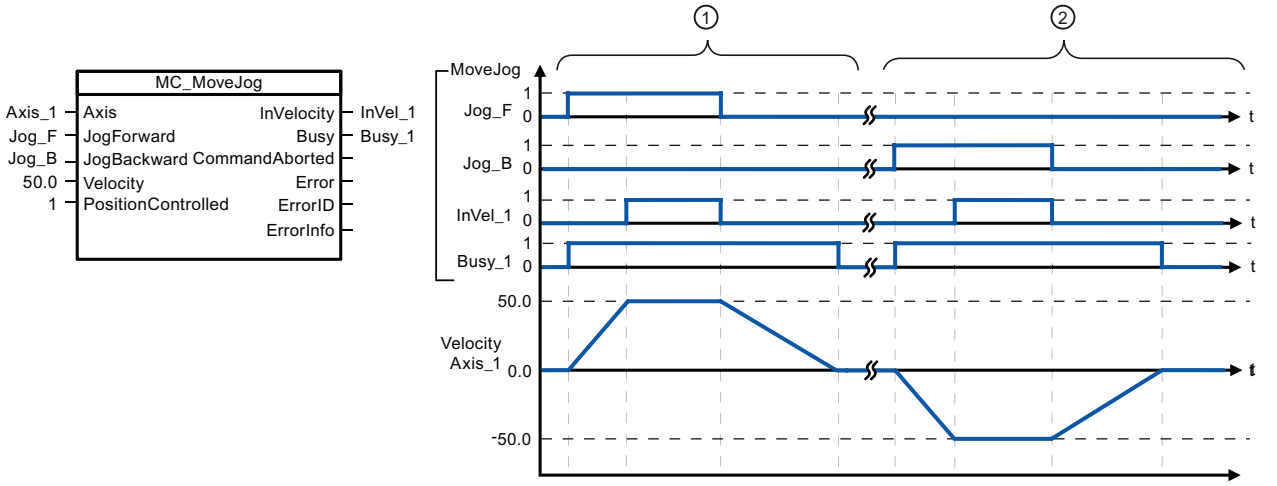
[2] В примере SCL "MC_MoveJog_DB" - это имя DB экземпляра.

Таблица 10-71 Параметры для инструкции MC_MoveJog

Параметр и тип данных		Тип данных	Описание
Axis	IN	TO_SpeedAxis	Технологический объект "Ось"
JogForward[1]	IN	Bool	При значении параметра TRUE, ось перемещается с указанной в параметре "Velocity" скоростью в положительном направлении. Знак перед значением параметра "Velocity" игнорируется. (значение по умолчанию: False)
JogBackward[1]	IN	Bool	При значении параметра TRUE, ось перемещается с указанной в параметре "Velocity" скоростью в отрицательном направлении. Знак перед значением параметра "Velocity" игнорируется. (значение по умолчанию: False)
Velocity	IN	Real	Предустановленная скорость для толчкового режима (значение по умолчанию: 100.0) Предельные значения: Стартстопная скорость ≤ |Velocity| ≤ Максимальная скорость
PositionControlled	IN	Bool	• 0: Управления по скорости • 1: Управление по положению (значение по умолчанию: true)
InVelocity	OUT	Bool	TRUE = Была достигнута скорость, указанная в параметре "Velocity".
Busy	OUT	Bool	TRUE = Задача выполняется.
CommandAborted	OUT	Bool	TRUE = При выполнении задача была отменена другой задачей.
Error	OUT	Bool	TRUE = Во время выполнения задачи возникла ошибка. Причина ошибки зафиксирована в параметрах ErrorID и ErrorInfo.

Параметр и тип данных		Тип данных	Описание
ErrorID	OUT	Word	ID ошибки для параметра "Error" (значение по умолчанию: 0000)
ErrorInfo	OUT	Word	ID информации об ошибке для параметра "ErrorID" (значение по умолчанию: 0000)

[1] Если оба параметра JogForward и JogBackward одновременно TRUE, то ось останавливается с заданным замедлением. Ошибка сигнализируется в параметрах "Error", "ErrorID" и "ErrorInfo".



Следующие значения были настроены в окне конфигурации "Динамика > Общее": Ускорение = 10,0 и замедление = 5,0

① Затем при помощи "Jog_F" ось перемещается в толчковом режиме в положительном направлении. Достижение осью новой заданной скорости 50,0 сигнализируется через "InVelo_1". Ось выполняет повторное торможение до состояния покоя после сброса Jog_F.

② Через "Jog_B" ось перемещается в толчковом режиме в отрицательном направлении. Достижение осью новой заданной скорости 50,0 сигнализируется через "InVelo_1". Ось выполняет повторное торможение до состояния покоя после сброса Jog_B.

Реакция на внешнее вмешательство

Задача MC_MoveJog может быть отменена следующими задачами управления перемещением:

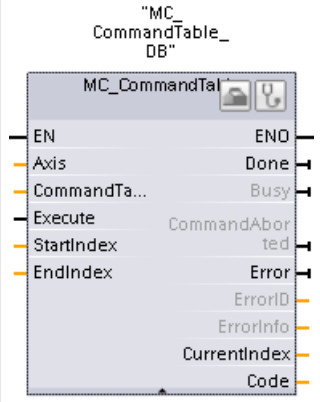
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Новая задача MC_MoveJog отменяет следующие активные задачи управления перемещением:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.3.7.10 Инструкция MC_CommandTable (выполнение команд управления осью как последовательности движений)

Таблица 10-72 Инструкция MC_CommandTable

LAD/FBD	SCL	Описание
	**"MC_CommandTable_DB"(** **Axis:=_multi_fb_in_,** **CommandTable:=_multi_fb_in_,** **Execute:=_bool_in_,** **StartIndex:=_uint_in_,** **EndIndex:=_uint_in_,** **Done=>_bool_out_,** **Busy=>_bool_out_,** **CommandAborted=>_bool_out_,** **Error=>_bool_out_,** **ErrorID=>_word_out_,** **ErrorInfo=>_word_out_,** **CurrentIndex=>_uint_out_,** **Code=>_word_out_);**	Выполняет серию отдельных движений для оси управления двигателем, которые могут объединяться в последовательность движений. Отдельные движения конфигурируются в таблице команд технологического объекты для формирования последовательности импульсов (TO_CommandTable_PTO).

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

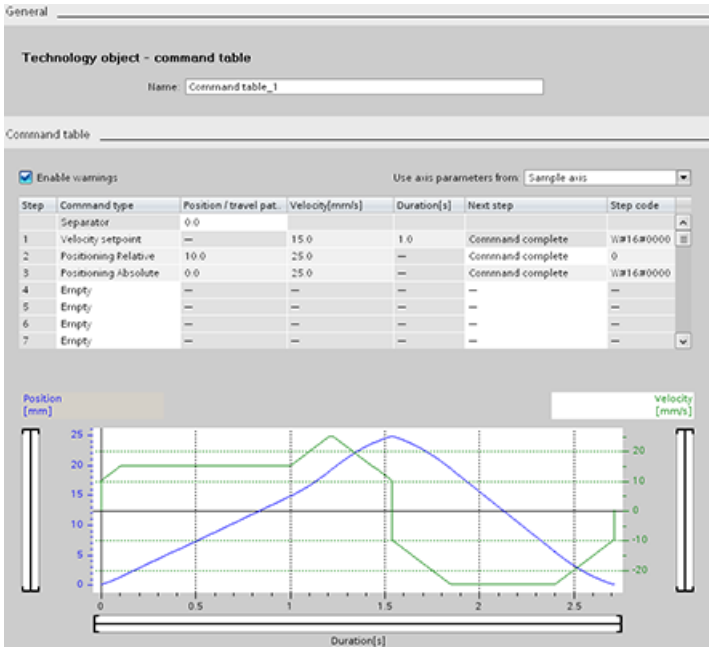
[2] В примере SCL "MC_CommandTable_DB" - это имя DB экземпляра.

Таблица 10-73 Параметры для инструкции MC_CommandTable

Параметр и тип данных		Тип данных	Начальное значение	Описание
Axis	IN	TO_Axis_1	-	Технологический объект "Ось"
Table	IN	TO_CommandTable_1	-	Технологический объект "Таблица команд"
Execute	IN	Bool	FALSE	Запуск задания по положительному фронту
StartIndex	IN	Int	1	Запуск обработки таблицы команд с этого шага Предельные значения: 1 ≤ **StartIndex** ≤ EndIndex
EndIndex	IN	Int	32	Завершение обработки таблицы команд с этого шага Предельные значения: StartIndex ≤ **EndIndex** ≤ 32
Done	OUT	Bool	FALSE	Обработка MC_CommandTable успешно завершена
Busy	OUT	Bool	FALSE	Операция выполняется
CommandAborted	OUT	Bool	FALSE	При выполнении задача была отменена другой задачей.
Error	OUT	Bool	FALSE	Ошибка при обработке. Причина определяется параметрами ErrorID и ErrorInfo..
ErrorID	OUT	Word	16#0000	ID ошибки
ErrorInfo	OUT	Word	16#0000	Информация об ошибке

Параметр и тип данных		Тип данных	Начальное значение	Описание
Step	OUT	Int	0	Шаг в настоящее время обрабатывается
Code	OUT	Word	16#0000	Пользовательский идентификатор находящегося в обработке шага

Требуемая последовательность движений может быть создана в окне конфигурации "Таблица команд", а для проверки результата можно использовать графическое представление на диаграмме изменений.



Можно выбирать типы команд, которые должны использоваться для обработки таблицы команд. Может быть введено до 32 заданий. Команды выполняются последовательно.

Таблица 10-74 Типы команд MC_CommandTable

Тип команды	Описание
Empty	Эта команда является заполнителем для любой добавляемой команды. Пустая запись игнорируется при обработке таблицы команд.
Halt	Команда останавливает ось. Примечание: Команда выполняется только после команды "Velocity setpoint".
Positioning Relative	Команда учитывает дистанцию при позиционировании оси. Она перемещает ось на указанное расстояние с указанной скоростью.
Positioning Absolute	Команда учитывает положение при позиционировании оси. Она перемещает ось на указанную позицию с заданной скоростью.
Velocity setpoint	Команда перемещает ось с указанной скоростью.

Тип команды	Описание
Wait	Команда ожидает, пока истечет указанный интервал времени. "Wait" не останавливает активное движение перемещения.
Separator	Команда добавляет разделительную линию ("Separator") над выбранной линией. Разделительная линия позволяет определить несколько профилей в одной таблице команд.

Обязательные условия для выполнения MC_CommandTable:

- Технологический объект TO_Axis_PTO V2.0 должен быть сконфигурирован правильно.
- Технологический объект TO_CommandTable_PTO должен быть сконфигурирован правильно
- Ось должна быть разблокирована.

Реакция на внешнее вмешательство

Задача MC_CommandTable может быть отменена следующими задачами управления перемещением:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog
- MC_CommandTable

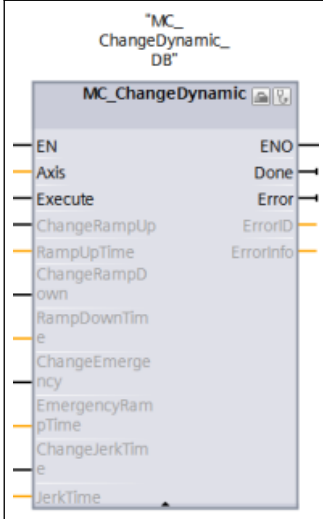
Новая задача MC_CommandTable отменяет следующие активные задачи управления перемещением:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog
- MC_CommandTable
- Текущее задание управления перемещением с запуском первой команды "Positioning Relative", "Positioning Absolute", "Velocity setpoint" или "Halt"

10.3.7.11 Инструкция MC_ChangeDynamic (изменение настроек динамики для оси)

Таблица 10-75 Инструкция MC_ChangeDynamic

LAD/FBD	SCL	Описание

	`"MC_ChangeDynamic_DB"(`
	`Execute:=_bool_in_,`
	`ChangeRampUp:=_bool_in_,`
	`RampUpTime:=_real_in_,`
	`ChangeRampDown:=_bool_in_,`
	`RampDownTime:=_real_in_,`
	`ChangeEmergency:=_bool_in_,`
	`EmergencyRampTime:=_real_in_,`
	`ChangeJerkTime:=_bool_in_,`
	`JerkTime:=_real_in_,`
	`Done=>_bool_out_,`
	`Error=>_bool_out_,`
	`ErrorID=>_word_out_,`
	`ErrorInfo=>_word_out_);`
		Изменяет динамические настройки оси управления перемещением:
		• Изменение значения для времени разгона (ускорение)
		• Изменение значения для времени торможения (замедление)
		• Изменение значения для времени аварийного торможения (аварийное замедление)
		• Изменение значения для времени сглаживания (рывок)

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "MC_ChangeDynamic_DB" - это имя DB экземпляра.

Таблица 10-76 Параметры для инструкции MC_ChangeDynamic

Параметр и тип данных		Тип данных	Описание
Axis	IN	TO_Axis_1	Технологический объект "Ось"
Execute	IN	Bool	Запуск команды по положительному фронту. Значение по умолчанию: FALSE
ChangeRampUp	IN	Bool	TRUE = Изменение времени разгона согласно входному параметру "RampUpTime". Значение по умолчанию: FALSE
RampUpTime	IN	Real	Время (в секундах) для разгона без ограничения рывка из состояния покоя до сконфигурированной максимальной скорости. Значение по умолчанию: 5,00 Изменение влияет на переменную (тег) <имя оси>. Config.DynamicDefaults.Acceleration. О действенности изменения можно узнать из описания этой переменной (тега).
ChangeRampDown	IN	Bool	TRUE = Изменение времени торможения согласно входному параметру "RampDownTime". Значение по умолчанию:FALSE
RampDownTime	IN	Real	Время (в секундах), за которое ось без ограничения рывка должна замедлиться от сконфигурированной максимальной скорости до состояния покоя. Значение по умолчанию: 5,00 Изменение влияет на переменную (тег) <имя оси>. Config.DynamicDefaults.Deceleration. О действенности изменения можно узнать из описания этой переменной (тега).

Параметр и тип данных		Тип данных	Описание
ChangeEmergency	IN	Bool	TRUE = Изменение времени аварийного торможения согласно входному параметру "EmergencyRampTime". Значение по умолчанию:FALSE
EmergencyRampTime	IN	Real	Время (в секундах), за которое ось в режиме аварийного отключения без ограничения рывка должна замедлиться от сконфигурированной максимальной скорости до состояния покоя. Значение по умолчанию: 2,00 Изменение влияет на переменную (тег) <имя оси>. Config.DynamicDefaults.EmergencyDeceleration. О действенности изменения можно узнать из описания этой переменной (тега).
ChangeJerkTime	IN	Bool	TRUE = Изменение времени сглаживания согласно входному параметру "JerkTime". Значение по умолчанию: FALSE
JerkTime	IN	Real	Время сглаживания (в секундах) для рампы разгона и торможения оси. Значение по умолчанию: 0,25 Изменение влияет на переменную (тег) <имя оси>. Config.DynamicDefaults.Jerk. О действенности изменения можно узнать из описания этой переменной (тега).
Done	OUT	Bool	TRUE = Измененные значения были записаны в технологический блок данных. Из описания этой переменной (тега) можно узнать вступили ли в силу изменения. Значение по умолчанию: FALSE
Error	OUT	Bool	TRUE = Во время выполнения команды возникла ошибка. Причина ошибки зафиксирована в параметрах ErrorID и ErrorInfo. Значение по умолчанию: FALSE
ErrorID	OUT	Word	ID ошибки. Значение по умолчанию: 16#0000
ErrorInfo	IN	Word	Информация об ошибке. Значение по умолчанию: 16#0000

Обязательные условия для выполнения MC_ ChangeDynamic:

- Технологический объект TO_Axis_PTO V2.0 должен быть сконфигурирован правильно.
- Ось должна быть разблокирована.

Примечание

Для подключения привода через PTO (выход шагового задания) можно использовать только инструкцию MC_ChangeDynamic.

Реакция на внешнее вмешательство

Команда MC_ChangeDynamic не может быть отменена никакой другой командой управления перемещением.

Новая команда MC_ChangeDynamic не отменяет никаких активных задач управления перемещением.

Примечание

Для входных параметров "RampUpTime", "RampDownTime", "EmergencyRampTime" и "RoundingOffTime" могут быть указаны значения, в результате чего параметры оси "Ускорение", "Замедление", "Аварийное замедление" и "Рывок" окажутся вне своих допустимых предельных значений.

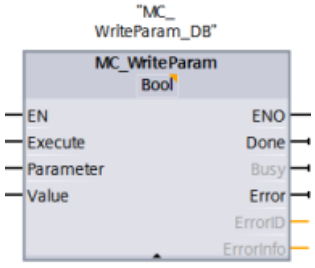
Убедиться, что параметры MC_ChangeDynamic не выходят из диапазона предельных значений для динамической конфигурации ТО "Ось".

10.3.7.12 Инструкция MC_WriteParam (запись параметров технологического объекта)

Инструкция MC_WriteParam записывает определенные параметры для изменения функциональности оси в программе пользователя.

Таблица 10-77 Инструкция MC_WriteParam

LAD/FBD	SCL	Описание

	`"MC_WriteParam_DB"(`
	`  Parameter:=_variant_in_,`
	`  Value:=_variant_in_,`
	`  Execute:=_bool_in_,`
	`  Done:=_bool_out_,`
	`  Error:=_real_out_,`
	`  ErrorID:=_word_out_,`
	`  ErrorInfo:=_word_out_);`
		С помощью инструкции MC_WriteParam возможна запись в открытые параметры (например, ускорение и значения в пользовательских DB).

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "MC_WriteParam_DB" - это имя DB экземпляра.

Возможность записи в открытые параметры. Запись в "MotionStatus" и "StatusBits" невозможна. В следующей таблице представлены действующие параметры:

Название параметра для записи	Название параметра для записи
Actor.InverseDirection	DynamicDefaults.Acceleration
Actor.DirectionMode	DynamicDefaults.Deceleration
Actor.DriveParameter.PulsesPerDriveRevolution	DynamicDefaults.Jerk
Sensor[1].ActiveHoming.Mode	DynamicDefaults.EmergencyDeceleration
Sensor[1].ActiveHoming.SideInput	PositionLimitsHW.Active
Sensor[1].ActiveHoming.Offset	PositionLimitsHW.MaxSwitchedLevel
Sensor[1].ActiveHoming.SwitchedLevel	PositionLimitsHW.MinSwitchedLevel
Sensor[1].PassiveHoming.Mode	PositionLimitsSW.Active
Sensor[1].PassiveHoming.SideInput	PositionLimitsSW.MinPosition
Sensor[1].PassiveHoming.SwitchedLevel	PositionLimitsSW.MaxPosition

Название параметра для записи	Название параметра для записи
Units.LengthUnit	Homing.AutoReversal
Mechanics.LeadScrew	Homing.ApproachDirection
DynamicLimits.MinVelocity	Homing.ApproachVelocity
DynamicLimits.MaxVelocity	Homing.ReferencingVelocity

Таблица 10-78 Параметры для инструкции MC_WriteParam

Параметр и тип данных		Тип данных	Описание
PARAMNAME	IN	Variant	Название параметра, в который будет записано значение
VALUE	IN	Variant	Значение, которое должно быть записано в назначенный параметр
EXECUTE	IN	Bool	Запуск инструкции. Значение по умолчанию: FALSE
DONE	OUT	Bool	Значение было записано. Значение по умолчанию: FALSE
BUSY	OUT	Bool	При TRUE инструкция выполняется в настоящий момент. Значение по умолчанию: FALSE
ERROR	OUT	Real	Если TRUE, значит произошла ошибка. Значение по умолчанию: FALSE
ERRORID	OUT	Word	ID ошибки
ERRORINFO	OUT	Word	Дополнительная информация для ERRORID

Таблица 10-79 Коды условий для ERRORID и ERRORINFO

ERRORID (W#16#...)	ERRORINFO (W#16#...)	Описание
0	0	Параметр TO-DB ось был успешно изменен
$8410_{[1]}$	$0028_{[1]}$	Недействительный параметр (осевой параметр TO-DB с неправильной длиной)
$8410_{[1]}$	$0029_{[1]}$	Недопустимый параметр (не осевой параметр TO-DB)
$8410_{[1]}$	$002B_{[1]}$	Недействительный параметр (осевой параметр TO-DB с защитой от записи)
$8410_{[1]}$	$002C_{[1]}$	Действительный параметр, но ось не выключена
Config $Error_{[2]}$	Config $Error_{[2]}$	Действительный параметр (открытый осевой параметр TO-DB с защитой от записи) в недопустимом диапазоне
Config $Error_{[3]}$	Config $Error_{[3]}$	Действительный параметр (открытый осевой параметр TO-DB) в недопустимом диапазоне

[1] Ошибка в MC_WriteParam

[2] Ошибка в MC_Power

[3] Ошибка в MC_Power и MC_MoveXXX или MC_CommandTable

Примечание

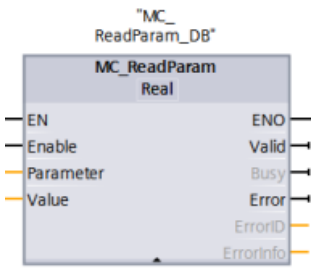
Для подключения привода через PROFIdrive-/аналоговый выход нельзя выполнять с MC_WriteParam запись параметров, для которых потребуется перезапуск технологического объекта. Информация в системе интерактивной помощи TIA Portal касательно этой инструкции не соответствует действительности.

10.3.7.13 Инструкция MC_ReadParam (чтение параметров технологического объекта)

Инструкция MC_ReadParam считывает определенные параметры, указывающие текущую позицию, скорость и т.п. определенной на входе "Axis" оси.

Таблица 10-80 Инструкция MC_ReadParam

LAD/FBD	SCL	Описание

	`"MC_ReadParam_DB"(`
	`Enable:=_bool_in_,`
	`Parameter:=_variant_in_,`
	`Value:=_variant_in_out_,`
	`Valid:=_bool_out_,`
	`Busy:=_bool_out_,`
	`Error:=_real_out_,`
	`ErrorID:=_word_out_,`
	`ErrorInfo:=_word_out_);`
		С помощью инструкции MC_ReadParam считываются отдельные значения состояния, независимо от контрольной точки в цикле.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

[2] В примере SCL "MC_ReadParam_DB" - это имя DB экземпляра.

Инструкция MC_ReadParam зависит от состояния "Enable". Пока вход "Enable" = true, инструкция считывает указанный "Параметр" в ячейку памяти "Value".

Значение "Position" переменной "MotionStatus" оси обновляется на основе текущего значения HSC в каждой контрольной точке цикла (CCP).

Значение "Velocity" переменной "MotionStatus" оси - это командная скорость в конце текущего сегмента (обновляется каждые ~10 мс). Инструкция MC_ReadParam может читать и это значение.

Если происходит ошибка, инструкция переходит в состояние ошибки, которое может быть сброшено только новым положительным фронтом на входе "Enable".

Таблица 10-81 Параметры для инструкции MC_ReadParam

Параметр и тип данных		Тип данных	Описание
ENABLE	IN	Bool	Запуск инструкции. Значение по умолчанию: FALSE
PARAMETER	IN	Variant	Указатель на считываемый ТО параметр
VALID	OUT	Bool	Если TRUE, значение было считано. Значение по умолчанию: FALSE

Параметр и тип данных		Тип данных	Описание
BUSY	OUT	Bool	Если TRUE, то инструкция обрабатывается. Значение по умолчанию: FALSE
ERROR	OUT	Real	Если TRUE, значит произошла ошибка. Значение по умолчанию: FALSE
ERRORID	OUT	Word	ID ошибки. Значение по умолчанию: 0
ERRORINFO	OUT	Word	Соответствующая информация о ERRORID. . Значение по умолчанию: 0
VALUE	INOUT	Variant	Указатель на место хранения считанного значения

Таблица 10-82 Коды условий для ERRORID и ERRORINFO

ERRORID (W#16#...)	ERRORINFO (W#16#...)	Описание
0	0	Параметр считан успешно
8410	0028	Недопустимый параметр (неправильная длина)
8410	0029	Недопустимый параметр (не TO-DB)
8410	0030	Недопустимый параметр (чтение невозможно)
8411	0032	Недопустимый параметр (неправильное значение)

Параметры TO

Переменная "MotionStatus" оси состоит из четырех значений. Возможен мониторинг изменений этих значений, которые могут быть считаны во время выполнения программы:

Имя переменной	Тип данных	Чтение через MC_ReadParam
MotionStatus:	Struktur	Нет
• Position (Позиция)	REAL	Да
• Velocity (Скорость)	REAL	Да
• Distance (Дистанция)	REAL	Да
• TargetPosition (Заданная позиция)	REAL	Да

10.3.8 Контроль активных команд

10.3.8.1 Контроль инструкции управления перемещением с помощью выходного параметра "Done"

Инструкции управления перемещением с выходным параметром "Done" запускаются посредством входного параметра "Execute" и имеют определенное завершение (например, для инструкции управления перемещением "MC_Home": реферирование выполнено успешно). Задача выполнена и ось в состоянии покоя.

- Выходной параметр "Done" сигнализирует значение TRUE, если задача успешно завершена.
- Выходные параметры "Busy", "CommandAborted" и "Error" сигнализируют, что задача все еще обрабатывается, была отменена, или имеется активная ошибка. Инструкцию управления перемещением "MC_Reset" невозможно отменить, и поэтому у нее нет выходного параметра "CommandAborted".
  - Во время обработки задачи управления перемещением выходной параметр "Busy" сигнализирует значение TRUE. Если задача была завершена, отменена или остановлена вследствие ошибки, выходной параметр "Busy" изменяет значение на FALSE. Это изменение происходит независимо от сигнала входного параметра "Execute".
  - Выходные параметры "Done", "CommandAborted" и "Error" сигнализируют значение TRUE по меньшей мере в течение одного цикла. Эти сообщения о состоянии сохраняются, пока входной параметр "Execute" остается на TRUE.

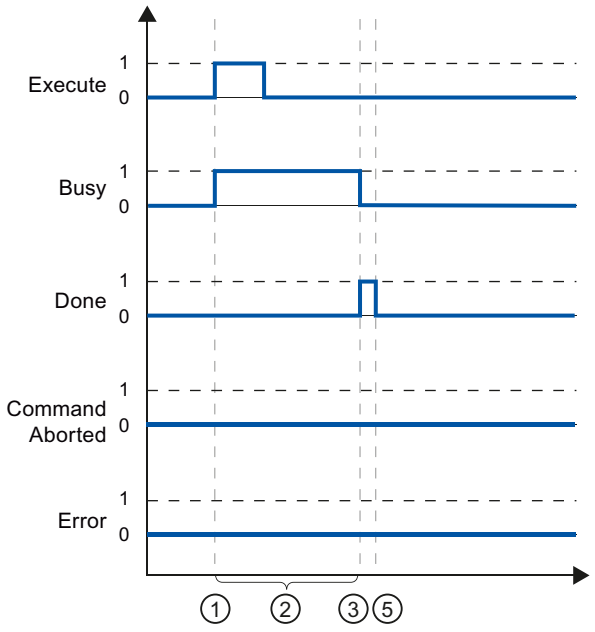
Задачи следующих инструкций управления перемещением имеют определенное завершение:

- MC_Reset
- MC_Home
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative

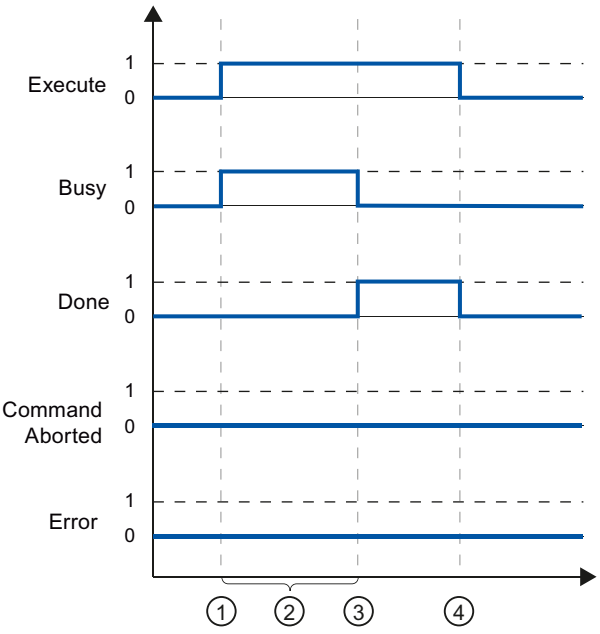
Поведение битов состояния представлено ниже для различных ситуаций в качестве примера.

- Первый пример показывает поведение оси для завершенной задачи. Если задача управления перемещением была полностью выполнена на момент ее завершения, это сигнализируется значением TRUE в выходном параметре "Done". Состояние сигнала входного параметра "Execute" влияет на длительность отображения в выходном параметре "Done".
- Второй пример показывает поведение оси для отмененной задачи. Если задача управления перемещением отменяется при выполнении, это сигнализируется значением TRUE в выходном параметре "CommandAborted". Состояние сигнала входного параметра "Execute" влияет на длительность отображения в выходном параметре "CommandAborted".
- Третий пример показывает поведение оси в случае ошибки. Если при выполнении задачи управления перемещением происходит ошибка, это сигнализируется значением TRUE в выходном параметре "Error". Состояние сигнала входного параметра "Execute" влияет на длительность отображения в выходном параметре "Error".

Таблица 10-83 Пример 1 - Полное выполнение задачи



Если "Execute" = FALSE при обработке задачи



Если "Execute" = FALSE после завершения задачи

① Задача запускается по положительному фронту на входном параметре "Execute". В зависимости от программирования, "Execute" может быть сброшен в FALSE во время выполнения задачи, или значение TRUE может быть удержано до завершения задачи.

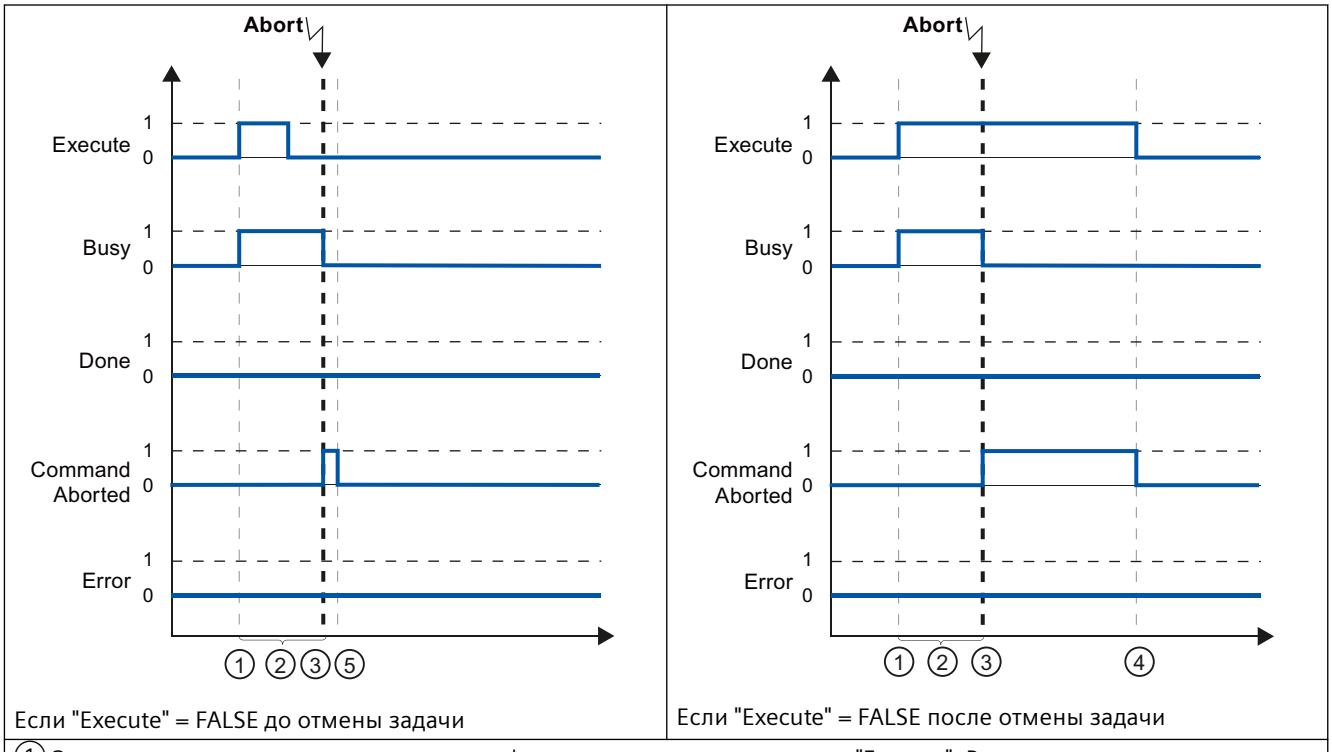
② При активной задаче выходной параметр "Busy" сигнализирует значение TRUE.

③ При завершении задачи (напр., инструкции управления перемещением "MC_Home": реферирование выполнено успешно), выходной параметр "Busy" изменяется на FALSE, а "Done" на TRUE.

④ Если "Execute" сохраняет значение TRUE до завершения задачи, то "Done" также остается TRUE и изменяет свое значение вместе с "Execute" на FALSE.

⑤ Если "Execute" был установлен на FALSE до завершения задачи, то "Done" сигнализирует значение TRUE только в течение одного цикла.

Таблица 10-84 Пример 2 - Отмена задачи



Если "Execute" = FALSE до отмены задачи

Если "Execute" = FALSE после отмены задачи

① Задача запускается по положительному фронту на входном параметре "Execute". В зависимости от программирования, "Execute" может быть сброшен в FALSE во время выполнения задачи, или значение TRUE может быть удержано до завершения задачи.

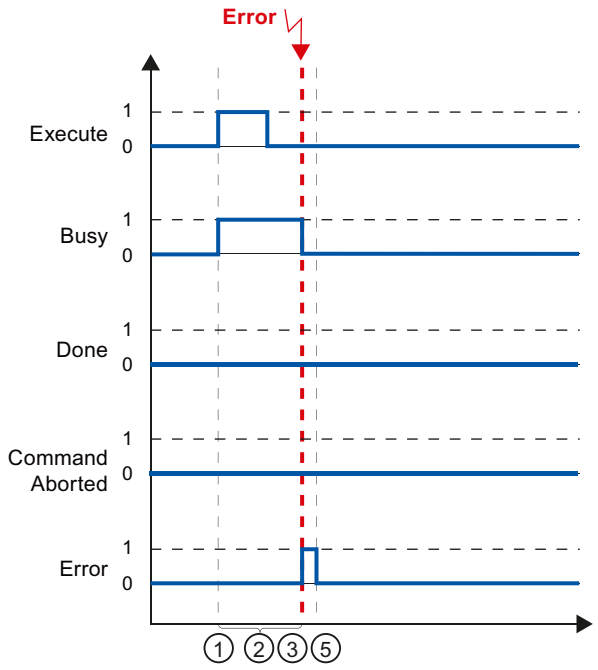
② При активной задаче выходной параметр "Busy" сигнализирует значение TRUE.

③ Задача отменяется при ее выполнении другой задачей управления перемещением. Если задача отменяется, выходной параметр "Busy" изменяется на FALSE, а "CommandAborted" на TRUE.

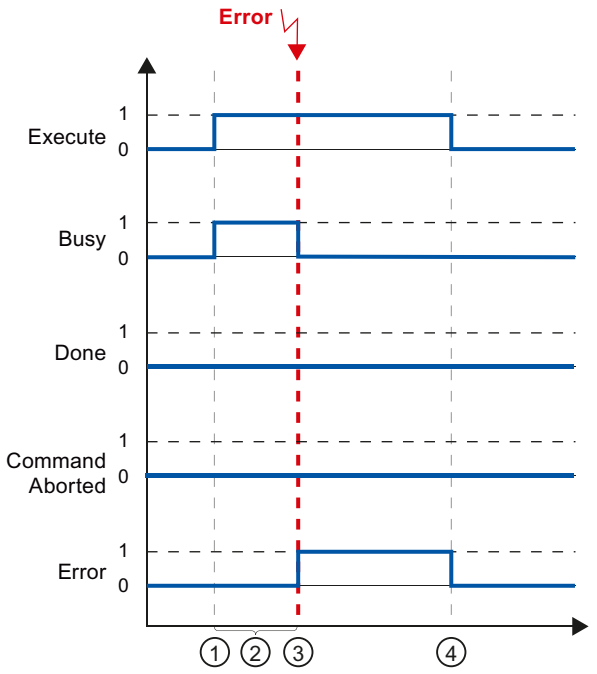
④ Если "Execute" сохраняет значение TRUE до отмены задачи, то "CommandAborted" также остается TRUE и изменяет свое значение вместе с "Execute" на FALSE.

⑤ Если "Execute" был установлен на FALSE до отмены задачи, то "CommandAborted" сигнализирует значение TRUE только в течение одного цикла.

Таблица 10-85 Пример 3 - Ошибка во время выполнения задачи



Если "Execute" = FALSE до возникновения ошибки



Если "Execute" = FALSE после возникновения ошибки

① Задача запускается по положительному фронту на входном параметре "Execute". В зависимости от программирования, "Execute" может быть сброшен в FALSE во время выполнения задачи, или значение TRUE может быть удержано до завершения задачи.

② При активной задаче выходной параметр "Busy" сигнализирует значение TRUE.

③ Во время выполнения задачи произошла ошибка. Если возникает ошибка, выходной параметр "Busy" изменяется на FALSE, а "Error" на TRUE.

④ Если "Execute" сохраняет значение TRUE до возникновения ошибки, то "Error" также остается TRUE и изменяет свое значение только вместе с "Execute" на FALSE.

⑤ Если "Execute" был установлен на FALSE до возникновения ошибки, то "Error" сигнализирует значение TRUE только в течение одного цикла.

10.3.8.2 Мониторинг MC_Velocity

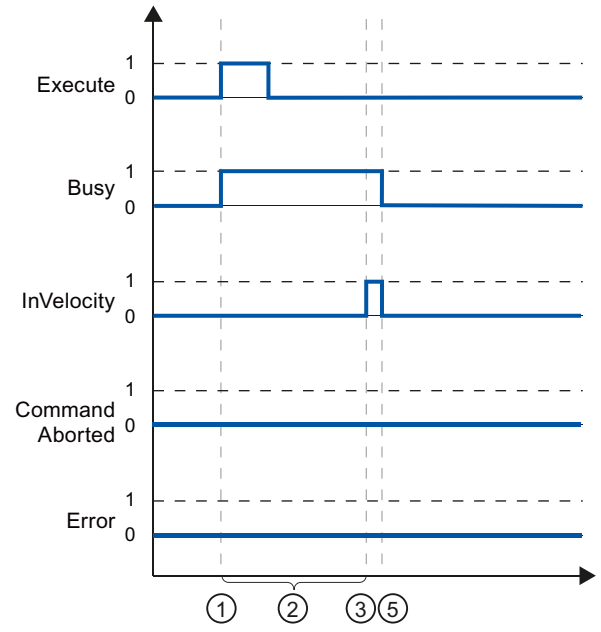
Задачи инструкции управления перемещением "MC_MoveVelocity" запускают перемещение с заданной скоростью.

- Задачи инструкции управления перемещением "MC_MoveVelocity" не имеют определенного завершения. Цель задачи достигнута в момент первого достижения заданной параметром скорости, и ось перемещается с постоянной скоростью. Когда заданная параметром скорость достигнута, это сигнализируется значением TRUE в выходном параметре "InVelocity".
- Задача завершена, если заданная скорость достигнута и входной параметр "Execute" принимает значение FALSE. Однако перемещение оси еще не завершено после завершения задачи. Пример: Перемещение оси может быть остановлено с помощью задачи управления перемещением "MC_Halt".
- Выходные параметры "Busy", "CommandAborted" и "Error" сигнализируют, что задача все еще обрабатывается, была отменена, или имеется активная ошибка.
  - Во время выполнения задачи управления перемещением выходной параметр "Busy" сигнализирует значение TRUE. Если задача была завершена, отменена или остановлена вследствие ошибки, выходной параметр "Busy" изменяет значение на FALSE. Это изменение происходит независимо от сигнала входного параметра "Execute".
  - Выходные параметры "InVelocity", "CommandAborted" и "Error" сигнализируют значение TRUE по меньшей мере в течение одного цикла, если их условия выполняются. Эти сообщения о состоянии сохраняются, пока входной параметр "Execute" остается на TRUE.

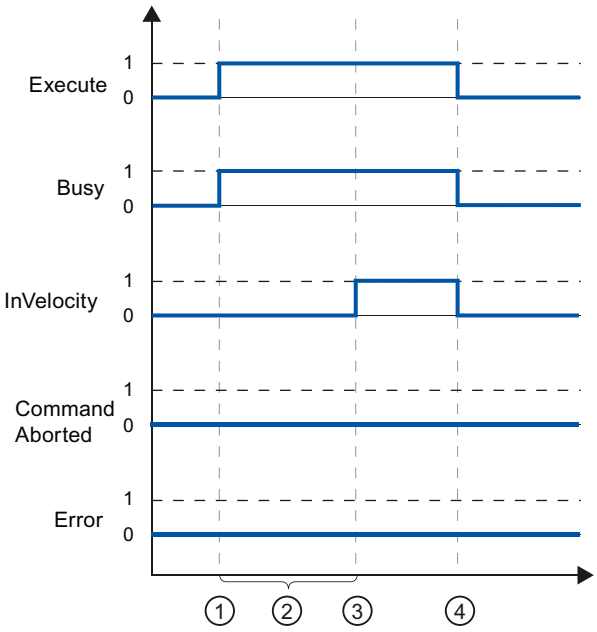
Поведение битов состояния представлено ниже для различных ситуаций в качестве примера:

- Первый пример демонстрирует поведение, когда ось достигает заданной параметром скорости. Если задача управления перемещением была выполнена к тому моменту, когда заданная параметром скорость достигнута, это сигнализируется значением TRUE в выходном параметре "InVelocity". Состояние сигнала входного параметра "Execute" влияет на длительность отображения в выходном параметре "InVelocity".
- Второй пример демонстрирует поведение, если задача отменяется до достижения заданной параметром скорости. Если задача управления перемещением отменяется до достижения заданной параметром скорости, это сигнализируется значением TRUE в выходном параметре "CommandAborted". Состояние сигнала входного параметра "Execute" влияет на длительность отображения в выходном параметре "CommandAborted".
- Третий пример демонстрирует поведение оси, если до достижения заданной параметром скорости возникает ошибка. Если при выполнении задачи управления перемещением до достижения заданной параметром скорости происходит ошибка, это сигнализируется значением TRUE в выходном параметре "Error". Состояние сигнала входного параметра "Execute" влияет на длительность отображения в выходном параметре "Error".

Таблица 10-86 Пример 1 – Если заданная параметром скорость достигнута



Если "Execute" = FALSE до достижения заданной параметром скорости



Если "Execute" = FALSE после достижения заданной параметром скорости

① Задача запускается по положительному фронту на входном параметре "Execute". В зависимости от особенностей программирования параметр "Execute" может снова принять значение FALSE еще до или только после достижения скорости, заданной в параметрах.

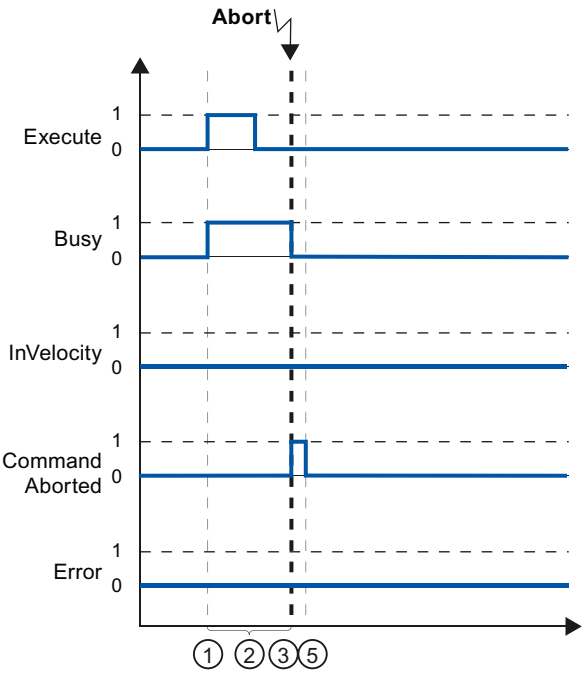
② При активной задаче выходной параметр "Busy" сигнализирует значение TRUE.

③ Если заданная скорость достигнута, выходной параметр "InVelocity" принимает значение TRUE.

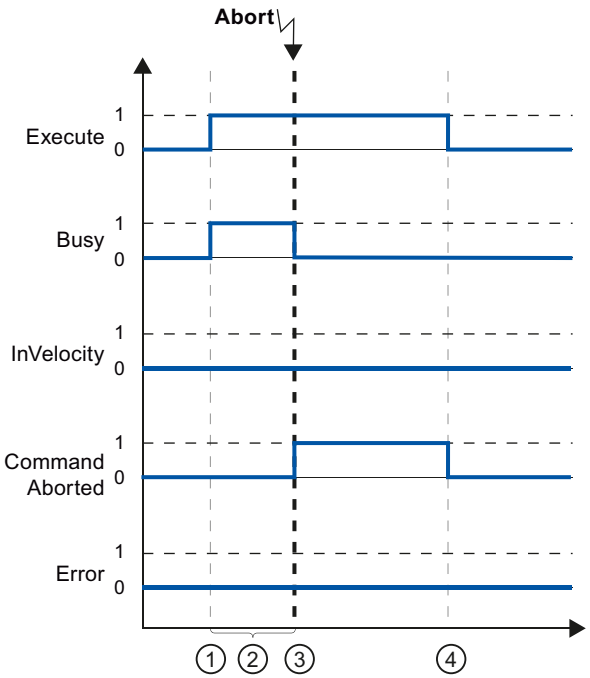
④ Если "Execute", сохраняет значение TRUE даже после того, как заданная параметром скорость была достигнута, задача остается активной. "InVelocity" и "Busy" сохраняют значение TRUE и их состояния изменяются в FALSE только вместе с "Execute".

⑤ Если "Execute" был сброшен в FALSE , прежде чем заданная параметром скорость будет достигнута, задача завершена, когда назначенная параметром скорость достигнута. InVelocity имеет значение TRUE на один цикл выполнения и изменяется в FALSE вместе с "Busy".

Таблица 10-87 Пример 2 - Если задача отменяется до достижения заданной параметром скорости



Если "Execute" = FALSE до отмены задачи



Если "Execute" = FALSE после отмены задачи

① Задача запускается по положительному фронту на входном параметре "Execute". В зависимости от программирования, "Execute" может быть сброшен в FALSE во время выполнения задачи, или значение TRUE может быть удержано до завершения задачи.

② При активной задаче выходной параметр "Busy" сигнализирует значение TRUE.

③ Задача отменяется при ее выполнении другой задачей управления перемещением. Если задача отменяется, выходной параметр "Busy" изменяется на FALSE, а "CommandAborted" на TRUE.

④ Если "Execute" сохраняет значение TRUE до отмены задачи, то ""CommandAborted" также остается TRUE и изменяет свое значение вместе с "Execute" на FALSE.

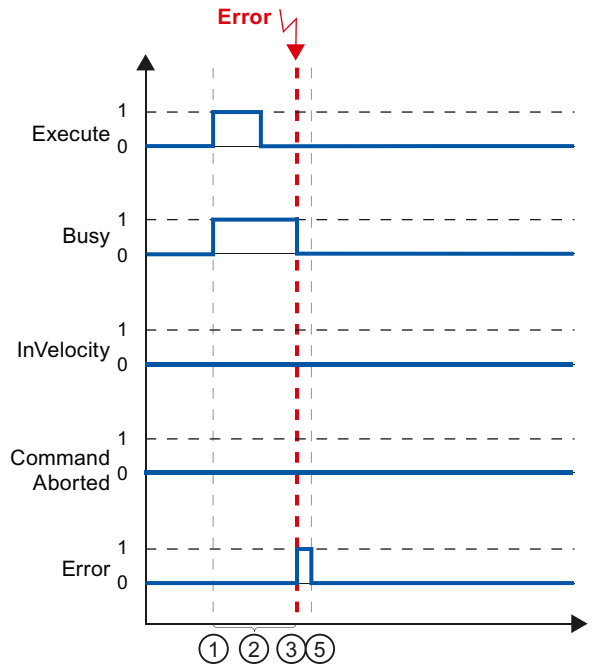
⑤ Если "Execute" был сброшен на FALSE до отмены задачи, то "CommandAborted" сигнализирует значение TRUE только в течение одного цикла.

Примечание

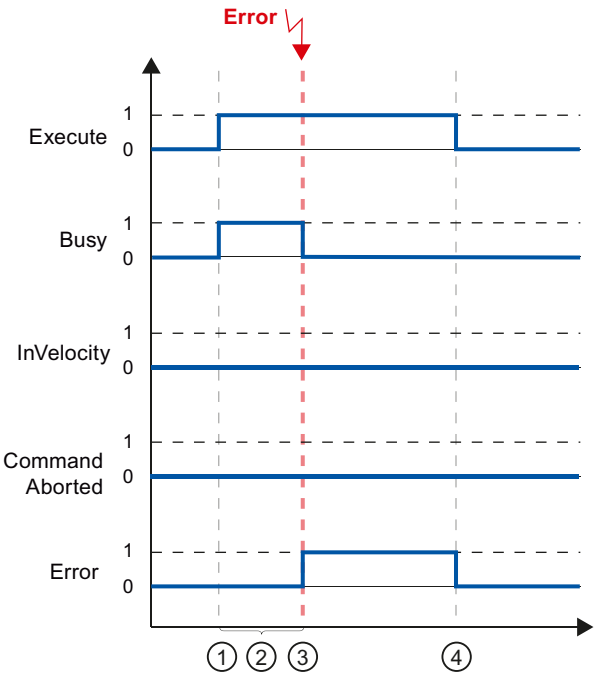
При следующих условиях отмена не отображается в выходном параметре "CommandAborted":

- Заданная параметром скорость была достигнута, входной параметр "Execute" имеет значение FALSE, и была запущена новая задача управления перемещением.
- Когда заданная параметром скорость была достигнута, и входной параметр "Execute" имеет значение FALSE, задача завершена. Поэтому запуск новой задачи не сигнализируется как отмена.

Таблица 10-88 Пример 3 - Если до достижения заданной параметром скорости возникает ошибка



Если "Execute" = FALSE до возникновения ошибки



Если "Execute" = FALSE после возникновения ошибки

① Задача запускается по положительному фронту на входном параметре "Execute". В зависимости от программирования, "Execute" может быть сброшен в FALSE во время выполнения задачи, или значение TRUE может быть удержано до завершения задачи.

② При активной задаче выходной параметр "Busy" сигнализирует значение TRUE.

③ Во время выполнения задачи произошла ошибка. Если возникает ошибка, выходной параметр "Busy" изменяется на FALSE, а "Error" на TRUE.

④ Если "Execute" сохраняет значение TRUE до возникновения ошибки, то "Error" также остается TRUE и изменяет свое значение только вместе с "Execute" на FALSE.

⑤ Если "Execute" был сброшен на FALSE до возникновения ошибки, то "Error" сигнализирует значение TRUE только в течение одного цикла.

Примечание

При следующих условиях ошибка не отображается в выходном параметре "Error":

- Заданная параметром скорость была достигнута, входной параметр "Execute" имеет значение FALSE, и возникает ошибка оси (например, наезд на программный конечный выключатель).
- Когда заданная параметром скорость была достигнута, и входной параметр "Execute" имеет значение FALSE, задача завершена. После завершения задачи ошибка оси сигнализируется только в инструкции управления перемещением "MC_Power".

10.3.8.3 Мониторинг инструкции MC_MoveJog

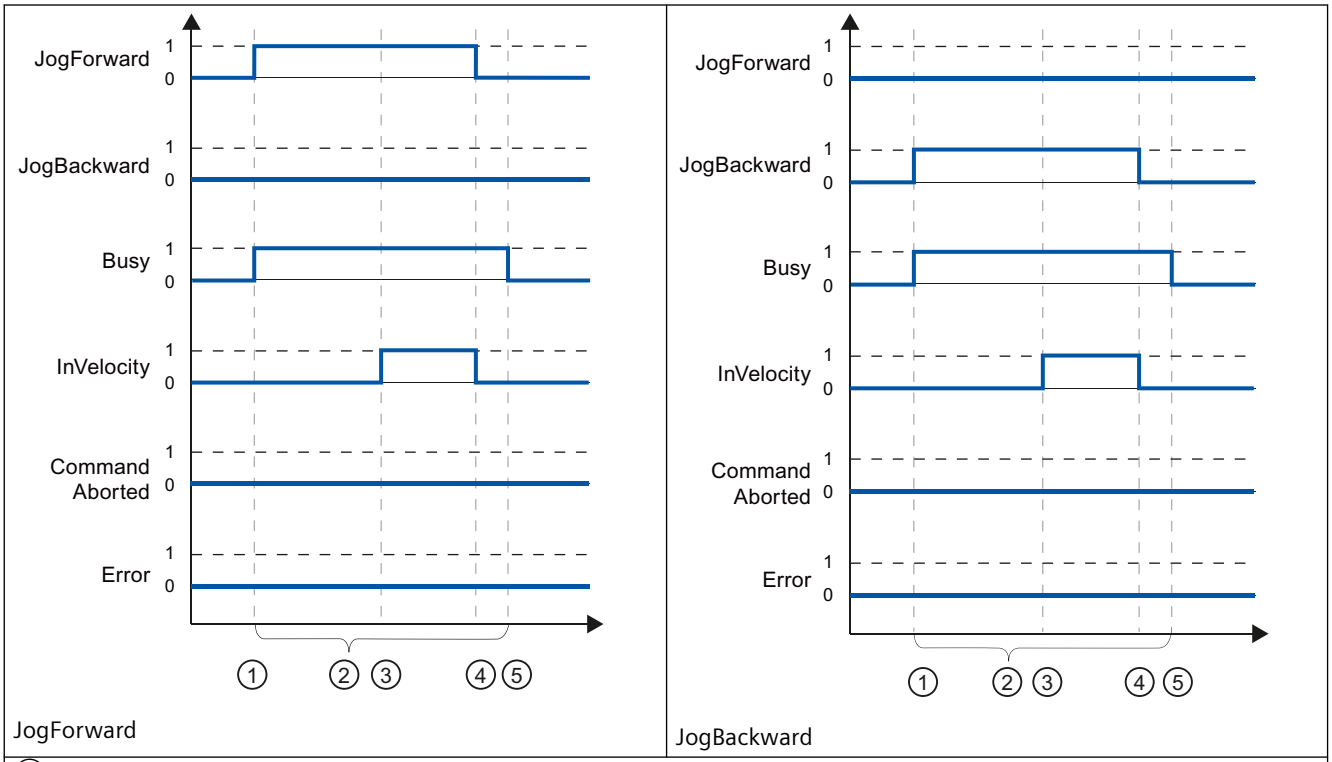
Задачи инструкции управления перемещением "MC_MoveJog" реализуют толчковый режим.

- Задачи управления перемещением "MC_MoveJog" не имеют определенного завершения. Цель задачи достигнута в момент первого достижения заданной параметром скорости, и ось перемещается с постоянной скоростью. Когда заданная параметром скорость достигнута, это сигнализируется значением TRUE в выходном параметре "InVelocity".
- Задача завершена, когда входной параметр "JogForward" или "JogBackward" был установлен в значение FALSE, и ось перешла в состояние покоя.
- Выходные параметры "Busy", "CommandAborted" и "Error" сигнализируют, что задача все еще обрабатывается, была отменена, или имеется активная ошибка.
  - Во время обработки задачи управления перемещением выходной параметр "Busy" сигнализирует значение TRUE. Если задача была завершена, отменена или остановлена вследствие ошибки, выходной параметр "Busy" изменяет значение на FALSE.
  - Выходной параметр "InVelocity" имеет состояние TRUE, пока ось перемещается с заданной параметром скоростью. Выходные параметры "", "CommandAborted" и "Error" сигнализируют состояние по меньшей мере в течение одного цикла. Эти сообщения о состоянии сохраняются, пока один из входных параметров "JogForward" или "JogBackward" остается на TRUE.

Поведение битов состояния представлено ниже для различных ситуаций в качестве примера.

- Первый пример демонстрирует поведение оси, когда заданная параметром скорость достигнута и поддерживается. Если задача управления перемещением была выполнена к тому моменту, когда заданная параметром скорость достигнута, это сигнализируется значением TRUE в выходном параметре "InVelocity".
- Второй пример демонстрирует поведение оси, если задача отменяется. Если задача управления перемещением отменяется при выполнении, это сигнализируется значением TRUE в выходном параметре "CommandAborted". Это происходит независимо от того, достигнута ли заданная скорость или нет.
- Третий пример показывает поведение оси в случае ошибки. Если при выполнении задачи управления перемещением происходит ошибка, это сигнализируется значением TRUE в выходном параметре "Error". Это происходит независимо от того, достигнута ли заданная скорость или нет.

Таблица 10-89 Пример 1 - Если заданная параметром скорость достигнута и поддерживается



① Задача запускается по положительному фронту на входном параметре "JogForward" или "JogBackward".

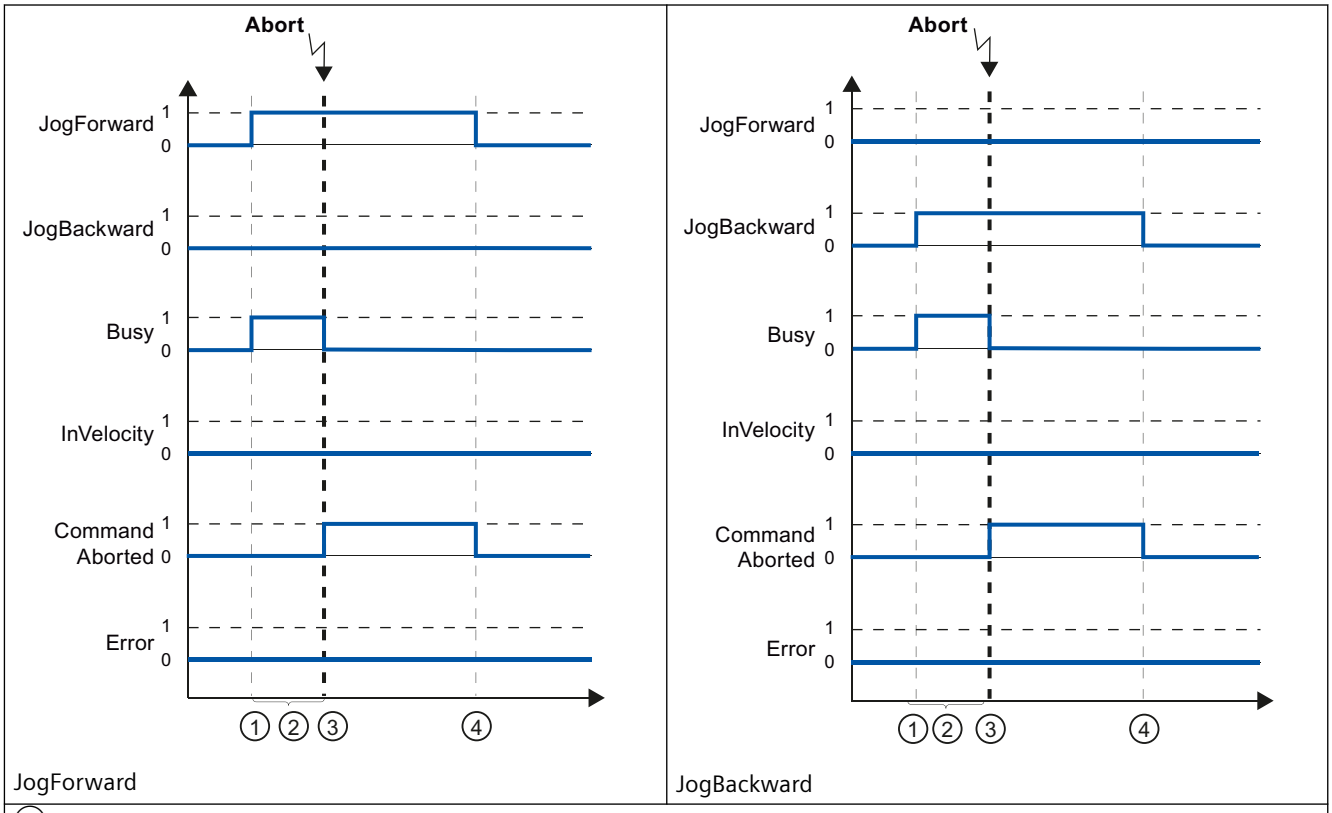
② При активной задаче выходной параметр "Busy" сигнализирует значение TRUE.

③ Если заданная скорость достигнута, выходной параметр "InVelocity" принимает значение TRUE.

④ Если входной параметр "JogForward" или "JogBackward" сбрасывается на значение FALSE, то движение оси завершается. Начинается замедление оси. Вследствие этого ось прекращает двигаться с постоянной скоростью и выходной параметр "InVelocity" становится FALSE.

⑤ Если ось перешла в состояние покоя, значит задача управления перемещением завершена и выходной параметр "Busy" становится FALSE.

Таблица 10-90 Пример 2 - Если задача отменяется при выполнении



① Задача запускается по положительному фронту на входном параметре "JogForward" или "JogBackward".

② При активной задаче выходной параметр "Busy" сигнализирует значение TRUE.

③ Задача отменяется при ее выполнении другой задачей управления перемещением. Если задача отменяется, выходной параметр "Busy" изменяется на FALSE, а "CommandAborted" на TRUE.

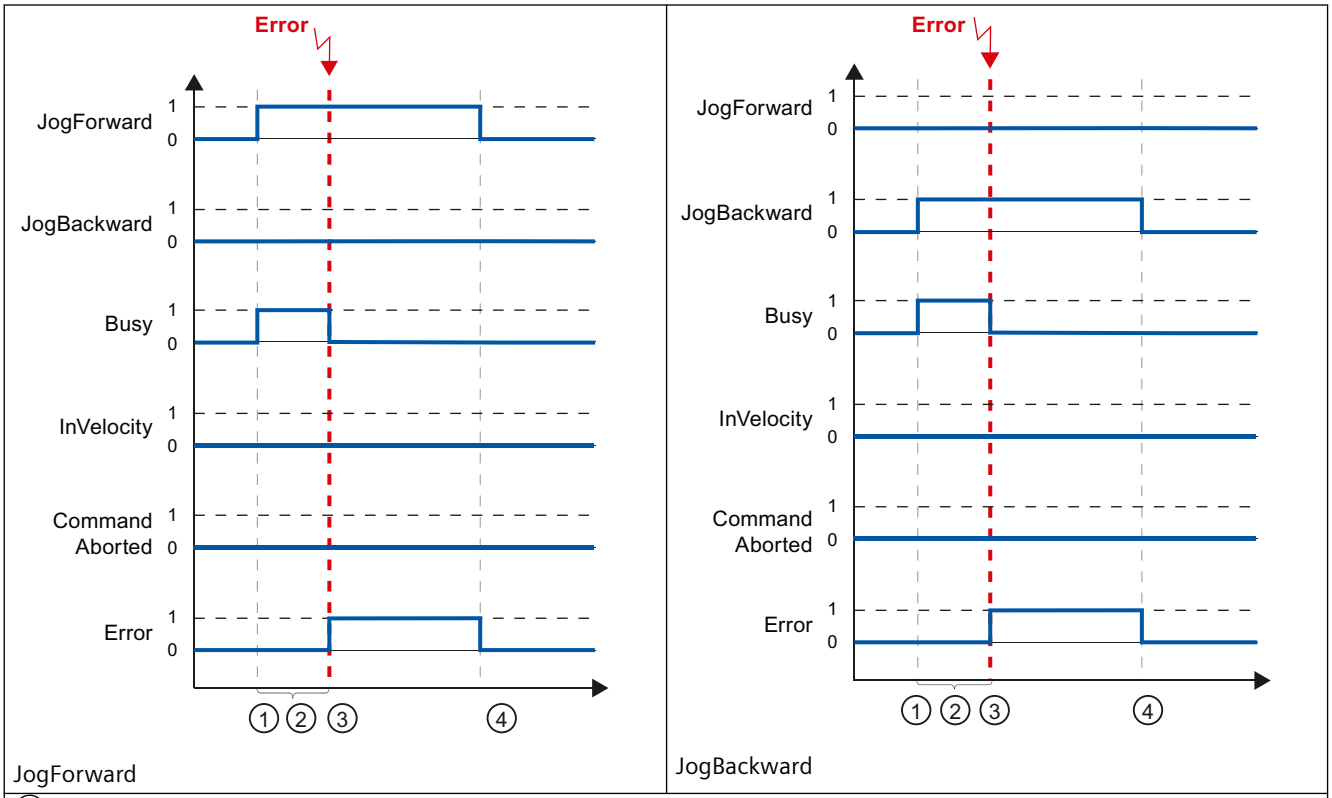
④ Если входной параметр "JogForward" или "JogBackward" сбрасывается на значение FALSE, то выходной параметр "CommandAborted" становится FALSE.

Примечание

Отмена задачи сигнализируется на выходном параметре "CommandAborted" только в течение одного цикла выполнения, если выполнены все нижеперечисленные условия:

На входных параметрах "JogForward" и "JogBackward" установилось значение FALSE (ось еще замедляется), и запускается новая инструкция управления перемещением.

Таблица 10-91 Пример 3 - Если во время выполнения задачи произошла ошибка



① Задача запускается по положительному фронту на входном параметре "JogForward" или "JogBackward".

② При активной задаче выходной параметр "Busy" сигнализирует значение TRUE.

③ Во время выполнения задачи произошла ошибка. Если возникает ошибка, выходной параметр "Busy" изменяется на FALSE, а "Error" на TRUE.

④ Если входной параметр "JogForward" или "JogBackward" сбрасывается на значение FALSE, то выходной параметр "Error" становится FALSE.

Примечание

Возникшая ошибка сигнализируется на выходном параметре "Error" только в течение одного цикла выполнения, если выполнены все нижеперечисленные условия:

На входных параметрах "JogForward" и "JogBackward" установилось значение FALSE (ось еще замедляется), и возникает новая ошибка (например, наезда на программный конечный выключатель).

10.3.9 Идентификаторы ошибок (ErrorID) и информационные сообщения об ошибках (ErrorInfo) для управления перемещением

В следующие таблице перечислены все ErrorID и ErrorInfo, которые могут встретиться в инструкциях управления перемещением и при переносе данных. Наряду с причиной ошибки, указываются и меры по устранению данной ошибки.

В зависимости от реакции на ошибку, ось при рабочих ошибках останавливает через осевой останов. Возможны следующие реакции на ошибку:

- **Отмена разрешения**
  Выдается заданное значение "нуль", и разрешение (деблокировка) отменяется. Ось в зависимости от конфигурации в приводе затормаживается и переводится в состояние покоя.
- **Останов по рампе аварийного останова**
  Активные команды управления перемещением отменяются. Ось выполняет торможение с аварийным замедлением без ограничения рывка, заданным в "Технологический объект > Расширенные параметры > Динамика > Рампа аварийного останова", и переходит в состояние покоя.

Рабочие ошибки с остановкой оси

ErrorID	ErrorInfo	Описание	Способ устранения	Реакция на ошибку
16#8000		**Ошибка привода, потеря "Привод готов"**		-
	16#0001	-	Квитировать ошибку при помощи инструкции "MC_Reset"; обеспечить сигнал привода; при необходимости перезапустить команду.	
16#8001		**Сработал нижний программный конечный выключатель**		Останов по рампе аварийного останова
	16#000E	Позиция нижнего программного конечного выключателя достигнута с текущим установленным замедлением.	Квитировать ошибку при помощи инструкции "MC_Reset"; следует использовать команду движения для отвода оси в положительном направлении из зоны программного конечного выключателя.	
	16#000F	Позиция нижнего программного конечного выключателя достигнута с аварийным замедлением.		
	16#0010	Позиция нижнего программного конечного выключателя нарушена с аварийным замедлением.		
16#8002		**Сработал верхний программный конечный выключатель**		Останов по рампе аварийного останова
	16#000E	Позиция верхнего программного конечного выключателя достигнута с текущим установленным замедлением.	Квитировать ошибку при помощи инструкции "MC_Reset"; следует использовать команду движения для отвода оси в отрицательном направлении из зоны программного конечного выключателя.	
	16#000F	Позиция верхнего программного конечного выключателя достигнута с аварийным замедлением.		
	16#0010	Позиция верхнего программного конечного выключателя нарушена с аварийным замедлением.		

ErrorID	ErrorInfo	Описание	Способ устранения	Реакция на ошибку
16#8003		**Достигнут нижний аппаратный конечный выключатель**		При подключении привода через PTO (Pulse Train Output): Останов по рампе аварийного останова При подключении привода через PROFIdrive: Отмена разрешения
	16#000E	Достигнут нижний аппаратный конечный выключатель. Ось была остановлена с аварийным замедлением. (При активном реферировании переключатель референтной точки не был найден.)	Квитировать ошибку при деблокированной оси с помощью инструкции "MC_Reset"; следует использовать команду движения для отвода оси в положительном направлении из зоны аппаратного конечного выключателя.	
16#8004		**Достигнут верхний аппаратный конечный выключатель**		При подключении привода через PTO (Pulse Train Output): Останов по рампе аварийного останова При подключении привода через PROFIdrive: Отмена разрешения
	16#000E	Достигнут верхний аппаратный конечный выключатель. Ось была остановлена с аварийным замедлением. (При активном реферировании переключатель референтной точки не был найден.)	Квитировать ошибку при деблокированной оси с помощью инструкции "MC_Reset"; следует использовать команду движения для отвода оси в отрицательном направлении из зоны аппаратного конечного выключателя.	
16#8005		**PTO/HSC уже используются другой осью**		-
	16#0001	-	**Ось была неправильно настроена:** Изменить конфигурацию PTO (Pulse Train Output) / HSC (High Speed Counter) и загрузить ее в контроллер.	
			Больше одной оси для PTO: Еще одна ось использует PTO или HSC. Если управление снова должно перейти к текущей оси, то другую ось необходимо отключить с помощью "MC_Power" Enable = FALSE.	
16#8006		**Ошибка обмена данными на панели оператора**		Отмена разрешения
	16#0012	Имеет место превышение времени.	Проверить подключение кабеля и повторно нажать клавишу для ручного управления.	
16#8007		**Ось не может быть разблокирована**		-
	16#0025	Перезапуск	Дождаться завершения перезапуска для оси.	
	16#0026	Выполняется загрузка в режиме RUN	Дождаться завершения процесса загрузки.	

ErrorID	ErrorInfo	Описание	Способ устранения	Реакция на ошибку
16#8008		**Недопустимое направление перемещения**		-
	16#002E	Выбранное направление перемещения является недопустимым.	• Изменить направление перемещения и повторить команду. • Установить допустимое направление вращения в конфигурации технологического объекта в "Расширенные параметры > Механика". Повторить команду.	
	16#002F	Движение в обратную сторону для выбранного направления перемещения невозможно.		
16#8009		**Нулевая метка переключателя референтной точки/энкодера не найдена**		Останов по рампе аварийного останова
	16#0033	Ошибка в конфигурации, оборудовании, при монтаже энкодера или на переключателе референтной точки.	• Подключить подходящее устройство. • Проверить устройство (ввод/вывод). • Сравнить конфигурирование HW-Config и технологического объекта.	
16#800A		**Аварийное сообщение от энкодера**		Отмена разрешения
	16#0001	-	Проверить функциональность устройства, соединения и ввод/вывод.	
	16#0034	Аппаратная ошибка на энкодере		
	16#0035	Энкодер загрязнен		
	16#0036	Ошибка при чтении абсолютного значения энкодера	Сравнить тип энкодера в приводе или в параметре энкодера P979 с данными конфигурации технологического объекта.	
	16#0037	Контроль нулевой метки энкодера	Энкодер сигнализирует ошибку при контроле нулевой метки (код ошибки 0x0002 в Gx_XIST2, см. профиль PROFIdrive). Проверить установку на предмет электромагнитной совместимости (ЭМС).	
	16#0038	Энкодер в состоянии парковки	• Выполнить поиск причины ошибки в подключенном приводе или энкодере. • Проверить, не появилось ли сообщение об ошибке вследствие процесса ввода в эксплуатацию на приводе или энкодере.	
	16#0040	PROFIdrive: Энкодер на шине вышел из строя (отказ станции)	Проверить функциональность устройства, соединения и ввод/вывод.	
	16#0041	PROFIdrive: Ошибка подтверждения работоспособности энкодера		

ErrorID	ErrorInfo	Описание	Способ устранения	Реакция на ошибку
16#800B		**Нарушение диапазона позиции**		Отмена разрешения
	16#0039	Нарушение диапазона в положительном направлении	Выполнить подвод оси к референтной точке в действительном диапазоне фактических значений.	
	16#003A	Нарушение диапазона в отрицательном направлении		
	16#003B	Изменение фактической позиции в сервотакте больше, чем модульная длина.	Настроить модульную длину используемого энкодера.	
16#800C		**Аварийное сообщение от привода**		Отмена разрешения
	16#0001	-	Проверить функциональность устройства, соединения и ввод/вывод.	
	16#003C	PROFIdrive: Сигнал привода "Запрос управления" пропал		
	16#003D	PROFIdrive: Привод был отключен.		
	16#003E	PROFIdrive: Привод на шине вышел из строя (отказ станции)		
	16#003F	PROFIdrive: Ошибка подтверждения работоспособности привода	• Проверить функциональность устройства, соединения и ввод/вывод. • Сравнить параметры такта HW-Config (линия PROFIBUS, Slave-OM для привода и энкодера) с исполнительной системой. Такт для Tmapc и Servo должен совпадать.	
16#800D		**Допустимая ошибка рассогласования была превышена**		Отмена разрешения
	16#0001	-	• Проверить конфигурацию контура регулирования. • Проверить сигнал направление энкодера. • Проверить конфигурацию контроля ошибки рассогласования.	
16#800E		**Ошибка на аппаратном конечном выключателе**		Отмена разрешения
	16#0042	Недопустимое свободное направление перемещения с активным аппаратным конечным выключателем	Запрограммированное направление движения деактивировано из-за активного аппаратного конечного выключателя. Выполнить отвод оси в противоположном направлении.	
	16#0043	Неправильная полярность аппаратного конечного выключателя, ось не может быть освобождена.	Проверить механическую конфигурацию аппаратного конечного выключателя.	
	16#0044	Активны оба аппаратных конечных выключателя, ось не может быть освобождена.		

ErrorID	ErrorInfo	Описание	Способ устранения	Реакция на ошибку
16#800F		**Ошибка в целевой области**		Отмена разрешения
	16#0045	Целевая область не достигнута	Целевая область не была достигнута в течение хронометрического допуска для позиционирования. • Проверить конфигурацию контроля позиционирования. • Проверить конфигурацию контура регулирования.	
	16#0046	Снова выполнен выход из целевой области	Произошел выход из целевой области в течение минимального времени ожидания. • Проверить конфигурацию контроля позиционирования. • Проверить конфигурацию контура регулирования.	
16#8010		**Позиция нижнего программного конечного выключателя превышает позицию верхнего программного конечного выключателя, если ось не является модульной осью**		Отмена разрешения
	16#0001	-	Изменить положение программных конечных выключателей.	

Рабочие ошибки без остановки оси

ErrorID	ErrorInfo	Описание	Способ устранения
16#8200		**Ось заблокирована**	
	16#0001	-	Разблокировать ось; повторить команду.
	16#003D	Отображается после отключения привода с аналоговым подсоединением привода.	Разблокировать ось; повторить команду.
16#8201		**Ось уже была разблокирована другим экземпляром "MC_Power"**	
	16#0001	-	Разблокировать ось только с одним экземпляром "MC_Power".
16#8202		**Максимальное число одновременных команд управления перемещением было превышено (макс. 200 команд для подсоединения привода через PTO (Pulse Train Output), макс. 100 команд для подсоединения привода через PROFIdrive/аналоговый выход)**	
	16#0001	-	Уменьшить число одновременно активных команд; повторить команду. Команда активна, если в инструкции управления перемещением параметр "Busy" = TRUE.
16#8203		**В настоящее время ось управляется в ручном режиме (панель управления осями)**	
	16#0001	-	Завершить ручное управление; повторить команду.
16#8204		**Ось не на референтной точке.**	
	16#0001	-	Переместить ось с помощью инструкции "MC_Home" на референтную точку; повторить команду.

ErrorID	ErrorInfo	Описание	Способ устранения
16#8205		**В данный момент ось управляется пользовательской программой (ошибка отображается только на панели управления осями)**	
	16#0013	Ось разблокирована в пользовательской программе.	Отключить ось инструкцией "MC_Power" и снова выбрать на панели управления осями ручное управление.
16#8206		**Технологический объект еще не активирован**	
	16#0001	-	Разблокировать ось с помощью Enable = TRUE инструкции "MC_Power" или активировать ось на панели управления осями.
16#8207		**Команда отклонена**	
	16#0016	Активное реферирование работает; нельзя запустить другой метод движения к референтной точке.	Ожидать завершения активного реферирования, или отменить активное реферирование с помощью команды перемещения, напр., "MC_Halt".
	16#0018	Ось не может перемещаться с помощью таблицы команд, если в данный момент выполняется ее активное или пассивное реферирование.	Ожидать завершения активного или пассивного реферирования.
	16#0019	Ось не может выполнять активное или пассивное реферирование, если в настоящий момент обрабатывается таблица команд.	Ожидать завершения таблицы команд, или отменить таблицу команд с помощью команды перемещения, напр., "MC_Halt".
	16#0052	Указанная позиция превышает числовое предельное значение.	Ввести в инструкции управления перемещением допустимое позиционное значение.
	16#0053	Ось разгоняется.	Ожидать готовности оси к работе.
	16#0054	Недействительное фактическое значение	Для выполнения команды "MC_Home" фактические значения должны быть действительными. Проверить состояние фактических значений. Переменная (тег) технологического объекта <имя оси>."StatusSensor.State" должна показывать значение 2 (действительно).
16#8208		**Недействительная разность между максимальной и стартстопной скоростью**	
	16#0002	Значение не является действительным числом.	Исправить значение; повторить команду.
	16#000A	Значение меньше или равно 0.	
16#8209		**Недействительное ускорение для технологического объекта "Ось"**	
	16#0002	Значение не является действительным числом.	Исправить значение; повторить команду.
	16#000A	Значение меньше или равно 0.	
16#820A		**Не удается перезапустить ось**	
	16#0013	Ось разблокирована в пользовательской программе.	Отключить ось с помощью инструкции "MC_Power"; выполнить перезапуск.
	16#0027	В настоящее время ось управляется в ручном режиме (панель управления осями)	Завершить режим ручного управления; выполнить перезапуск.
	16#002C	Ось не выключена.	Выключить ось. Повторить команду.
	16#0047	Технологический объект не готов для перезапуска.	Заново загрузить проект в целевую систему.
	16#0048	Условие для перезапуска технологического объекта не выполнено.	Выключить технологический объект.

ErrorID	ErrorInfo	Описание	Способ устранения
16#820B		**Таблица команд не может быть выполнена**	
	16#0026	Выполняется загрузка в режиме RUN	Дождаться завершения процесса загрузки.
16#820C		**Конфигурация отсутствует**	
	16#0001	-	Внутренняя ошибка Связаться с горячей линией.

Ошибка параметра блока

ErrorID	ErrorInfo	Описание	Способ устранения
16#8400		**Недействительное значение в параметре "Position" инструкции управления перемещением**	
	16#0002	Значение не является действительным числом.	Исправить значение; повторить команду.
	16#0005	Значение находится за пределами диапазона чисел (больше, чем 1E+12)	
	16#0006	Значение находится за пределами диапазона чисел (меньше, чем 1E+12)	
16#8401		**Недействительное значение в параметре "Distance" инструкции управления перемещением**	
	16#0002	Значение не является действительным числом.	Исправить значение; повторить команду.
	16#0005	Значение находится за пределами диапазона чисел (больше, чем 1E+12)	
	16#0006	Значение находится за пределами диапазона чисел (меньше, чем 1E+12)	
16#8402		**Недействительное значение в параметре "Velocity" инструкции управления перемещением**	
	16#0002	Значение не является действительным числом.	Исправить значение; повторить команду.
	16#0008	Значение превышает максимальную настроенную скорость	
	16#0009	Значение меньше сконфигурированной стартстопной скорости	
	16#0024	Значение меньше 0	
16#8403		**Недействительное значение в параметре "Direction" инструкции управления перемещением**	
	16#0011	Значение выбора недействительно.	Исправить значение выбора; повторить команду.
16#8404		**Недействительное значение в параметре "Mode" инструкции управления перемещением**	

ErrorID	ErrorInfo	Описание	Способ устранения
	16#0011	Значение выбора недействительно.	Исправить значение выбора; повторить команду.
	16#0015	Не выполнено конфигурирование активного или пассивного реферирования	Изменить конфигурацию и загрузить ее в контроллер; деблокировать ось и подать команду заново.
	16#0017	Смена направления была активирована на аппаратном конечном выключателе, хотя аппаратные конечные выключатели деактивированы.	• Активировать аппаратный конечный выключатель с помощью переменной (тега) <имя оси>.PositionLimitsHW.Active = TRUE, повторить команду. • Изменить конфигурацию и загрузить ее в контроллер; деблокировать ось и подать команду заново.
	16#0055	Недействительный режим на инкрементальном энкодере	Запустить движение к референтной точке для инкрементального энкодера через параметр "Mode" = 0, 1, 2, 3.
	16#0056	Недействительный режим на абсолютном энкодере	Пассивное и активное реферирование ("Mode" = 2, 3) невозможно для абсолютного энкодера. Запустить движение к референтной точке для абсолютного энкодера через параметр "Mode" = 0, 1.
16#8405		**Недействительное значение в параметре "StopMode" инструкции управления перемещением**	
	16#0011	Значение выбора недействительно.	Исправить значение выбора; снова разблокировать ось.
16#8406		**Одновременная толчковая подача в прямом и обратном направлении запрещена**	
	16#0001	-	Убедиться, что параметры "JogForward" и "JogBackward" не имею одновременного состояния сигнала TRUE; повторить команду.
16#8407		**Переключение на другую ось с помощью инструкции "MC_Power" возможно только после отключения активной оси**	
	16#0001	-	Отключить активную ось; после можно переключиться на другую ось и разблокировать ее.
16#8408		**Недействительное значение в параметре "Axis" инструкции управления перемещением**	
	16#001A	Указанное значение не соответствует требуемой версии технологического объекта.	Исправить значение; повторить команду.
	16#001B	Указанное значение не соответствует требуемому типу технологического объекта.	
	16#001C	Указанное значение не является технологическим блоком данных для управления перемещением	
16#8409		**Недействительное значение в параметре "CommandTable" инструкции управления перемещением**	
	16#001A	Указанное значение не соответствует требуемой версии технологического объекта.	Исправить значение; повторить команду.
	16#001B	Указанное значение не соответствует требуемому типу технологического объекта.	
	16#001C	Указанное значение не является технологическим блоком данных для управления перемещением	

ErrorID	ErrorInfo	Описание	Способ устранения
16#840A		**Недействительное значение в параметре "StartStep" инструкции управления перемещением**	
	16#000A	Значение меньше или равно 0.	Исправить значение; повторить команду.
	16#001D	Начальный шаг больше конечного шага.	
	16#001E	Значение больше 32	
16#840B		**Недействительное значение в параметре "EndStep" инструкции управления перемещением**	
	16#000A	Значение меньше или равно 0.	Исправить значение; повторить команду.
	16#001E	Значение больше 32	
16#840C		**Недействительное значение в параметре "RampUpTime" инструкции управления перемещением**	
	16#0002	Значение не является действительным числом.	Исправить значение; повторить команду.
	16#000A	Значение меньше или равно 0.	
16#840D		**Недействительное значение в параметре "RampDownTime" инструкции управления перемещением**	
	16#0002	Значение не является действительным числом.	Исправить значение; повторить команду.
	16#000A	Значение меньше или равно 0.	
16#840E		**Недействительное значение в параметре "EmergencyRampTime" инструкции управления перемещением**	
	16#0002	Значение не является действительным числом.	Исправить значение; повторить команду.
	16#000A	Значение меньше или равно 0.	
16#840F		**Недействительное значение в параметре "JerkTime" инструкции управления перемещением**	
	16#0002	Значение не является действительным числом.	Исправить значение; повторить команду.
	16#000A	Значение меньше или равно 0.	
16#8410		**Недействительное значение в параметре "Parameter" инструкции управления перемещением**	
	16#0002	Значение не является действительным числом.	Исправить значение; повторить команду.
	16#000B	Адрес недействителен	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
	16#0028	Тип данных VARIANT-указателя "Параметр" и "Значение" не совпадают.	Использовать подходящий тип данных; повторить команду.
	16#0029	VARIANT-указатель "Параметр" ссылается не на блок данных технологического объекта.	Исправить VARIANT-указатель; повторить команду.
	16#002A	Невозможно прочесть значение в VARIANT-указателе "Параметр".	Исправить VARIANT-указатель; повторить команду.
	16#002B	Невозможно записать значение в VARIANT-указателе "Параметр".	Исправить VARIANT-указатель или значение; повторить команду.
	16#002C	Ось не выключена.	Выключить ось. Повторить команду.
16#8411		**Недействительное значение в параметре "Value" инструкции управления перемещением**	

ErrorID	ErrorInfo	Описание	Способ устранения
	16#0002	Значение не является действительным числом.	Исправить значение; повторить команду.
	16#0005	Значение находится за пределами диапазона чисел (больше, чем 1E+12)	
	16#0006	Значение находится за пределами диапазона чисел (меньше, чем 1E+12)	

Ошибка конфигурации оси

ErrorID	ErrorInfo	Описание	Способ устранения
16#8600		**Недопустимое параметрирование генератора импульсов (PTO)**	
	16#000B	Адрес недействителен.	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
	16#0014	Выбранное оборудование используется другим приложением.	
16#8601		**Недопустимое параметрирование быстрого счетчика (HSC)**	
	16#000B	Адрес недействителен.	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
	16#0014	Выбранное оборудование используется другим приложением.	
16#8602		**Недействительное параметрирование "Разрешающего выхода"**	
	16#000B	Адрес недействителен.	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
16#8603		**Недействительное параметрирование "Входа готовности"**	
	16#000B	Адрес недействителен.	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
16#8604		**Недействительное значение "Импульсы на оборот двигателя"**	
	16#000A	Значение меньше или равно нулю	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
16#8605		**Недействительное значение "Дистанция за оборот"**	
	16#0002	Значение не является действительным числом.	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
	16#0005	Значение находится за пределами диапазона чисел (больше, чем 1E+12)	
	16#000A	Значение меньше или равно нулю	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8606		**Недействительное значение "Стартстопная скорость"**	

ErrorID	ErrorInfo	Описание	Способ устранения
	16#0002	Значение не является действительным числом.	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
	16#0003	Значение превышает верхний аппаратный предел	
	16#0004	Значение меньше нижнего аппаратного предела	
	16#0007	Стартстопная скорость превышает максимальную скорость.	
16#8607		**Недействительное значение "Максимальная скорость"**	
	16#0002	Значение не является действительным числом.	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
	16#0003	Значение превышает верхний аппаратный предел	
	16#0004	Значение меньше нижнего аппаратного предела	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8608		**Недействительное значение "Ускорение"**	
	16#0002	Значение не является действительным числом.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
	16#0003	Значение превышает верхний аппаратный предел	
	16#0004	Значение меньше нижнего аппаратного предела	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8609		**Недействительное значение "Замедление"**	

ErrorID	ErrorInfo	Описание	Способ устранения
	16#0002	Значение не является действительным числом.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
	16#0003	Значение превышает верхний аппаратный предел	
	16#0004	Значение меньше нижнего аппаратного предела	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#860A		**Недействительное значение "Аварийное замедление"**	
	16#0002	Значение не является действительным числом.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
	16#0003	Значение превышает верхний аппаратный предел	
	16#0004	Значение меньше нижнего аппаратного предела	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#860B		**Значение позиции для нижнего программного конечного выключателя недействительно**	
	16#0002	Значение не является действительным числом.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
	16#0005	Значение находится за пределами диапазона чисел (больше, чем 1E+12)	
	16#0006	Значение находится за пределами диапазона чисел (меньше, чем 1E+12)	
	16#0030	Значение позиции нижнего программного конечного выключателя больше значения позиции верхнего программного конечного выключателя.	
16#860C		**Значение для позиции верхнего программного конечного выключателя недействительно**	

ErrorID	ErrorInfo	Описание	Способ устранения
	16#0002	Значение не является действительным числом.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
	16#0005	Значение находится за пределами диапазона чисел (больше, чем 1E+12)	
	16#0006	Значение находится за пределами диапазона чисел (меньше, чем 1E+12)	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#860D		**Адрес нижнего аппаратного конечного выключателя недействителен**	
	16#000B	Недействительный адрес	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
	16#000C	Адрес отрицательного фронта недействителен.	
	16#000D	Адрес положительного фронта недействителен.	
16#860E		**Адрес верхнего аппаратного конечного выключателя недействителен**	
	16#000B	Недействительный адрес	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
	16#000C	Адрес отрицательного фронта недействителен.	
	16#000D	Адрес положительного фронта недействителен.	
16#860F		**Недействительное значение "Смещение референтной точки"**	
	16#0002	Значение не является действительным числом.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
	16#0005	Значение находится за пределами диапазона чисел (больше, чем 1E+12)	
	16#0006	Значение находится за пределами диапазона чисел (меньше, чем 1E+12)	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8610		**Недействительное значение "Скорость подвода"**	

ErrorID	ErrorInfo	Описание	Способ устранения
	16#0002	Значение не является действительным числом.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
	16#0008	Скорость выше максимальной скорости.	
	16#0009	Скорость ниже стартстопной скорости.	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8611		**Недействительное значение "Скорость движения к референтной точке"**	
	16#0002	Значение не является действительным числом.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
	16#0008	Скорость выше максимальной скорости.	
	16#0009	Скорость ниже стартстопной скорости.	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8612		**Недействительный адрес переключателя референтной точки**	
	16#000B	Недействительный адрес	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
	16#000C	Адрес отрицательного фронта недействителен.	
	16#000D	Адрес положительного фронта недействителен.	
16#8613		**При активном движении к референтной точке смена направления была активирована на аппаратном конечном выключателе, хотя аппаратные конечные выключатели не активированы.**	
	16#0001	-	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8614		**Недействительное значение "Рывок"**	

ErrorID	ErrorInfo	Описание	Способ устранения
	16#0002	Значение не является действительным числом.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
	16#001F	Значение превышает максимальный рывок	
	16#0020	Значение меньше минимально рывка	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8615		**Значение для "Единицы измерения" недействительно**	
	16#0011	Значение выбора недействительно.	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
16#8616		**Адрес переключателя референтной точки недействителен (пассивное реферирование от версии 4)**	
	16#0011	Значение выбора недействительно.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8617		**Значение переменной (тега) <имя оси>.Sensor.Sensor[1].ActiveHoming.Mode недействительно**	
	16#0011	Значение выбора недействительно. (Допустимое значение: 2 = Реферирование через цифровой вход)	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8618		**Значение переменной (тега) <имя оси>.Sensor.Sensor[1].PassiveHoming.Mode недействительно**	

ErrorID	ErrorInfo	Описание	Способ устранения
	16#0011	Значение выбора недействительно. (Допустимое значение: 2 = Реферирование через цифровой вход)	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8619		**Значение переменной (тега) <имя оси>.Actor.Type недействительно**	
	16#0011	Значение выбора недействительно. (Допустимое значение: 2 = Соединение через импульсный интерфейс)	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#861A		**Значение для "Допустимого направления вращения" недействительно**	
	16#0011	Значение выбора недействительно.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
	16#002D	"Оба направления" недопустимо, если выход направления деактивирован.	
16#861B		**Ошибочные передаточные числа силового редуктора**	
	16#0031	Недействительное значение.	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
16#861C		**Недопустимая комбинация данных для движения к референтной точке с инкрементным энкодером**	
	16#0031	Недействительное значение.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#861D		**Установлен неправильный способ монтажа энкодера. Недействительное значение в <имя оси>.Sensor.Sensor[1].MountingMode**.	
	16#0011	Значение выбора недействительно.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#861E		**Недействительная конфигурация окружности у измерительного колесика энкодера. Недействительное значение в <имя оси>.Sensor.Sensor[1].Parameter.DistancePerRevolution**.	

ErrorID	ErrorInfo	Описание	Способ устранения
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#861F		**Ошибочная конфигурация для разрешения линейного энкодера. Недействительное значение в <имя оси>.Sensor.Sensor[1].Parameter.Resolution**.	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8620		**Установленное точное разрешение для Gn_XIST1 недействительно. Недействительное значение в <имя оси>.Sensor.Sensor[1].Parameter.FineResolutionXist1**.	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8621		**Установленное точное разрешение для Gn_XIST1 в <имя оси>.Sensor.Sensor[1].Parameter.FineResolutionXist1 не совпадает с установкой в PROFIdrive параметре P979**.	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8622		**Недействительное значение для конфигурационных данных <имя оси>.Actor.Interface.AddressIn или <имя оси>.Actor.Interface.AddressOut**	
	16#0011	Значение выбора недействительно.	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
16#8623		**Определенное в переменной (теге) <имя оси>.Sensor.Sensor[1].Type значение недействительно.**	
	16#0011	Значение выбора недействительно.	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
16#8624		**Определенная система датчика недействительна. Недействительное значение в <имя оси>.Sensor.Sensor[1].System**.	

ErrorID	ErrorInfo	Описание	Способ устранения
	16#0011	Значение выбора недействительно.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8625		**Ошибка параметра позиционного контроля. Недействительное значение в <имя оси>.PositioningMonitoring.MinDwellTime**.	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8626		**Ошибка параметра позиционного контроля. Недействительное значение в <имя оси>.PositioningMonitoring.Window**.	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8627		**Ошибка конфигурации интерфейса PROFIdrive фактического значения. Недействительное значение в <имя оси>.Sensor.Sensor[1].Interface.AddressIn или <имя оси>.Sensor.Sensor[1].Interface.AddressOut**	
	16#0011	Значение выбора недействительно.	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
16#8628		**Неправильные коэффициенты управления**	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	Неправильное значение для усиления или предуправления для контура регулирования. • Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду. (<имя оси>.PositionControl.Kv, <имя оси>.PositionControl.Kpc)
16#8629		**Предельное значение для сигнала состояния покоя ошибочное. Недействительное значение в <имя оси>.StandStillSignal.VelocityThreshold**.	

ErrorID	ErrorInfo	Описание	Способ устранения
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#862A		**Ошибка параметра позиционного контроля. Недействительное значение в <имя оси>.PositioningMonitoring.ToleranceTime**.	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#862B		**Противоречивое параметрирование PROFIBUS; сумма Ti и To больше цикла DP**	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
16#862C		**Ошибка параметра контроля состояния покоя. Недействительное значение в <имя оси>.StandStillSignal.MinDwellTime**.	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#862D		**Ошибка параметра контроля ошибки рассогласования. Недействительное значение в <имя оси>.FollowingError.MinValue**.	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#862E		**Недействительное значение для конфигурационных данных <имя оси>.Modulo.Length**	

ErrorID	ErrorInfo	Описание	Способ устранения
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#862F		**Недействительное значение для конфигурационных данных <имя оси>.Modulo.StartValue**	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8630		**Недействительное значение для конфигурационных данных <имя оси>.Actor.DriveParameter.ReferenceSpeed**	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8631		**Установленное точное разрешение для Gn_XIST2 недействительно. Недействительное значение в <имя оси>.Sensor.Sensor[1].Parameter.FineResolutionXist2**.	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8632		**Количество определяемых оборотов энкодера недействительно. Недействительное значение в <имя оси>.Sensor.Sensor[1].Parameter.DeterminableRevolutions**.	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8633		**Указанное направление подвода для переключателя референтной точки для пассивного реферирования недействительно. Недействительное значение в <имя оси>.Sensor.Sensor[1].PassiveHoming.Direction**.	

ErrorID	ErrorInfo	Описание	Способ устранения
16#8634		**Ошибка параметра контроля ошибки рассогласования. Недействительное значение в <имя оси>.FollowingError.MaxValue.**	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8635		**Ошибка параметра контроля ошибки рассогласования. Недействительное значение в <имя оси>.FollowingError.MinVelocity.**	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8636		**Неправильный коэффициент управления. Недействительное значение для коэффициента предуправления <имя оси>.PositionControl.Kpc**	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
16#8637		**Недействительное значение для конфигурационных данных <имя оси>.Sensor.Sensor[1].Interface.Type**	
	16#0011	Значение выбора недействительно.	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
16#8638		**Недействительное значение для конфигурационных данных <имя оси>.Sensor.Sensor[1].Interface.HSC**	
	16#0011	Значение выбора недействительно.	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
16#8639		**Ошибка в приводе**	
	16#0049	Ошибка конфигурации в устройстве	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
	16#004A	Сервотакт слишком большой для технологии.	Внутренняя системная ошибка. Проверить проект на предмет целостности и заново загрузить его в контроллер.
	16#004B	Драйвер устройства не был инициализирован при загрузке.	Для активации технологического объекта, драйвер исполнительного устройства должен быть инициализирован. Повторить команду позднее.

ErrorID	ErrorInfo	Описание	Способ устранения
16#863A		**Ошибка при обмене данными с приводом**	
	16#004C	Ошибка конфигурации в устройстве	• Подключить подходящее устройство. • Проверить устройство (ввод/вывод). • Сравнить конфигурирование HW-Config и технологического объекта.
	16#004D	Сервотакт слишком большой для драйвера устройства.	• Подключить подходящее устройство. • Проверить устройство (ввод/вывод). • Сравнить конфигурирование HW-Config и технологического объекта.
	16#004E	Ошибка внутреннего обмена данными с устройством	Проверить проект на предмет целостности и заново загрузить его в контроллер.
16#863B		**Ошибка в энкодере**	
	16#0049	Ошибка конфигурации в устройстве	Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось.
	16#004A	Сервотакт слишком большой для технологии.	Внутренняя системная ошибка. Проверить проект на предмет целостности и заново загрузить его в контроллер.
	16#004B	Драйвер устройства не был инициализирован при загрузке.	Для активации технологического объекта, драйвер исполнительного устройства должен быть инициализирован. Повторить команду позднее.
16#863C		**Ошибка при обмене данными с энкодером**	
	16#004C	Ошибка конфигурации в устройстве	• Подключить подходящее устройство. • Проверить устройство (ввод/вывод). • Сравнить конфигурирование HW-Config и технологического объекта.
	16#004D	Сервотакт слишком большой для драйвера устройства.	• Подключить подходящее устройство. • Проверить устройство (ввод/вывод). • Сравнить конфигурирование HW-Config и технологического объекта.
	16#004E	Ошибка внутреннего обмена данными с устройством	Проверить проект на предмет целостности и заново загрузить его в контроллер.
16#863D		**Ошибка при обмене данными с устройством (приводом или энкодером)**	

ErrorID	ErrorInfo	Описание	Способ устранения
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн; квитировать ошибку при помощи инструкции "MC_Reset" и при необходимости перезапустить команду.
	16#0055	Запрошенный логический адрес недействителен.	• Подключить подходящее устройство. • Проверить устройство (ввод/вывод). • Проверить топологическую конфигурацию в HW-Config. • Сравнить конфигурирование HW-Config и технологического объекта.
	16#0056	Запрошенный логический адрес вывода недействителен.	
	16#0057	Запрошенный логический адрес вывода недействителен.	
16#863E		**Значение переменной (тега) "ControlPanel.Input.TimeOut" ist недействительно (панель управления осями)**	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	Исправить значение в переменной (теге) технологического объекта <имя оси>.ControlPanel.Input.TimeOut). Указать значение в миллисекундах (мс).
16#863F		**Недействительное значение для конфигурационных данных <имя оси>.Actor.DriveParameter.MaxSpeed**	
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	Исправить опорное значение в приводе и в конфигурации ТО в Actuator.MaxSpeed/2. При аналоговом подключении привода исправить опорное значение в приводе и в конфигурации ТО в Actuator.MaxSpeed/1.17.

Ошибки при переносе данных

ErrorID	ErrorInfo	Описание	Способ устранения	Реакция на ошибку
16#8640		**Ошибка при переносе конфигурации привода**		
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	Перезапуск	Ошибка при переносе
	16#0059	Устройство не согласовано с модулем привода SINAMICS или не поддерживает необходимые для переноса службы.	Перезапуск	Ошибка при переносе
	16#005A	Перенос данных был отменен из-за недостатка ресурсов.	Перезапуск	Ошибка при переносе
	16#005B	Перенос данных возможен только при прямом подключении устройства к области I/O.	Перезапуск	Ошибка при переносе

ErrorID	ErrorInfo	Описание	Способ устранения	Реакция на ошибку
	16#005C	Максимальная скорость (p1082): Параметр отсутствует, его значение не может быть считано или оно выходит за пределы разрешенного диапазона. Считывание параметров было отменено из-за сигнализации оборудованием ошибки.	Перезапуск	Ошибка при переносе
	16#005D	Макс. вращающий момент/сила (p1520): Параметр отсутствует, его значение не может быть считано или оно выходит за пределы разрешенного диапазона. Считывание параметров было отменено из-за сигнализации оборудованием ошибки.	Перезапуск	Ошибка при переносе
	16#005E	Макс. вращающий момент/сила (p1521): Параметр отсутствует, его значение не может быть считано или оно выходит за пределы разрешенного диапазона. Считывание параметров было отменено из-за сигнализации оборудованием ошибки.	Перезапуск	Ошибка при переносе
	16#005F	Точное разрешение вращающего момента/предельное значение силы (p1544): Параметр отсутствует, его значение не может быть считано или оно выходит за пределы разрешенного диапазона. Считывание параметров было отменено из-за сигнализации оборудованием ошибки.	Перезапуск	Ошибка при переносе
	16#0060	Основная скорость/ном. скорость (p2000): Параметр отсутствует, его значение не может быть считано или оно выходит за пределы разрешенного диапазона. Считывание параметров было отменено из-за сигнализации оборудованием ошибки.	Перезапуск	Ошибка при переносе
	16#0061	Ном. вращающий момент/ном. сила (p2003): Параметр отсутствует, его значение не может быть считано или оно выходит за пределы разрешенного диапазона. Считывание параметров было отменено из-за сигнализации оборудованием ошибки.	Перезапуск	Ошибка при переносе
16#8041		**Ошибка при переносе конфигурации энкодера**		
	16#0030	Значение имеет неправильный числовой формат или находится вне допустимого числового диапазона	Перезапуск	Ошибка при переносе

ErrorID	ErrorInfo	Описание	Способ устранения	Реакция на ошибку
	16#005A	Перенос данных был отменен из-за недостатка ресурсов.	Перезапуск	Ошибка при переносе
	16#005B	Перенос данных возможен только при прямом подключении устройства к области I/O.	Перезапуск	Ошибка при переносе
	16#0059	Устройство не согласовано с модулем привода SINAMICS или не поддерживает необходимые для переноса службы.	Перезапуск	Ошибка при переносе
	16#0062	Система датчика (r0979[1/11].0): Параметр отсутствует, его значение не может быть считано или оно выходит за пределы разрешенного диапазона. Считывание параметров было отменено из-за сигнализации оборудованием ошибки.	Перезапуск	Ошибка при переносе
	16#0063	Разрешение энкодера (r0979[2/12]): Параметр отсутствует, его значение не может быть считано или оно выходит за пределы разрешенного диапазона. Считывание параметров было отменено из-за сигнализации оборудованием ошибки.	Перезапуск	Ошибка при переносе
	16#0064	Разрешение энкодера Gx_XIST1 (r0979[3/13]): Параметр отсутствует, его значение не может быть считано или оно выходит за пределы разрешенного диапазона. Считывание параметров было отменено из-за сигнализации оборудованием ошибки.	Перезапуск	Ошибка при переносе
	16#0065	Разрешение энкодера Gx_XIST2 (r0979[4/14]): Параметр отсутствует, его значение не может быть считано или оно выходит за пределы разрешенного диапазона. Считывание параметров было отменено из-за сигнализации оборудованием ошибки.	Перезапуск	Ошибка при переносе
	16#0066	Количество разрешаемых оборотов энкодера (r0979[5/15]): Параметр отсутствует, его значение не может быть считано или оно выходит за пределы разрешенного диапазона. Считывание параметров было отменено из-за сигнализации оборудованием ошибки.	Перезапуск	Ошибка при переносе
16#8642		**Внутренний перенос конфигурации**		

ErrorID	ErrorInfo	Описание	Способ устранения	Реакция на ошибку
	16#0067	1: Недопустимое значение для Actor.MaxSpeed (Actor.MaxSpeed больше, чем 2*Actor.ReferenceSpeed); способ устранения: К примеру, в приводе установлено P2000 = P1082.	Сброс	Ошибка конфигурации
16#8643		**Расхождения между ТО и конфигурацией привода**		
	16#0068	Настроенный тип телеграммы несовместим с типом телеграммы устройства (P922 или P2079).	Сброс	Ошибка конфигурации
	16#0069	Несовместимое разрешение вращающего момента.	Сброс	Ошибка конфигурации
	16#006A	Элементарная продолжительность периода для цикла мастер-приложения не совпадает с таковой для серво-такта.	Сброс	Ошибка конфигурации
	16#006B	Такт обработки ТО не совпадает с прикладным циклом привода.	Сброс	Ошибка конфигурации
	16#006C	В приводе установлен функциональный модуль с линейным двигателем.	Сброс	Ошибка конфигурации
16#8644		**Расхождения между ТО и конфигурацией энкодера**		
	16#0068	Настроенный тип телеграммы несовместим с типом телеграммы устройства (P922 или P2079).	Сброс	Ошибка конфигурации
	16#006A	Элементарная продолжительность периода для цикла мастер-приложения не совпадает с таковой для серво-такта.	Сброс	Ошибка конфигурации
	16#006B	Такт обработки ТО не совпадает с прикладным циклом привода.	Сброс	Ошибка конфигурации
	16#006D	Датчик на приводе не является абсолютным энкодером (P979).	Сброс	Ошибка конфигурации
16#8645		**Максимальная скорость с приводом и установленными параметрами оси не может быть достигнута**		
	16#0001	Общее	Сброс	Ошибка конфигурации

Ошибка конфигурации таблицы команд

ErrorID	ErrorInfo	Описание	Способ устранения
16#8700		**Значение для "Типа команды" в таблице команд недействительно**	
	16#0001	-	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн и при необходимости перезапустить команду.

ErrorID	ErrorInfo	Описание	Способ устранения
16#8701		**Значение для "Позиции/траектории перемещения" в таблице команд недействительно**	
	16#0002	Значение не является действительным числом.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн и при необходимости перезапустить команду.
	16#0005	Значение находится за пределами диапазона чисел (больше, чем 1E+12)	
	16#0006	Значение находится за пределами диапазона чисел (меньше, чем 1E+12)	
16#8702		**Значение для "Скорости" в таблице команд недействительно**	
	16#0002	Значение не является действительным числом.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн и при необходимости перезапустить команду.
	16#0008	Значение превышает максимальную настроенную скорость	
	16#0009	Значение меньше сконфигурированной стартстопной скорости	
16#8703		**Значение для "Продолжительности" в таблице команд недействительно**	
	16#0002	Значение не является действительным числом.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн и при необходимости перезапустить команду.
	16#0021	Значение больше 64800 с	
	16#0022	Значение меньше 0,001 с	
16#8704		**Значение для "Следующего шага" в таблице команд недействительно**	
	16#0011	Значение выбора недействительно.	• Загрузить правильную конфигурацию в контроллер; с помощью инструкции "MC_Power" заново разблокировать ось. • Исправить неправильное значение онлайн и при необходимости перезапустить команду.
	16#0023	Смена команды для этой команды недопустима.	

Внутренние ошибки

ErrorID	ErrorInfo	Описание	Способ устранения
16#8FFF		**Внутренняя ошибка**	
	16#F0**	-	Выключить и снова включить CPU. Если после этого ошибка остается, обратиться в службу поддержки для клиентов. При обращении потребуется следующая информация: • ErrorID • ErrorInfo • Записи в буфере диагностики

Коммуникация

11

Устройство S7-1200 предлагает несколько типов коммуникаций между CPU и программаторами, устройствами HMI и другими CPU.

⚠ ПРЕДУПРЕЖДЕНИЕ

Если злоумышленник сможет получить физический доступ к сетям пользователя, то он сможет читать и записывать данные.

TIA Portal, CPU и устройства HMI (кроме устройств HMI, работающих с GET / PUT) используют безопасную связь с защитой от взломов путём замещения оригинала и атак через посредника. После активации коммуникации, обмен подписанными сообщениями происходит в форме открытого текста, который позволяет злоумышленнику читать данные, но защищает от их несанкционированной записи. TIA Portal, а не процесс коммуникации, шифрует данные блоков с защитой ноу-хау.

Все другие формы коммуникации (обмен I/O данным через PROFIBUS, PROFINET, AS-i или другую шину ввода-вывода, GET / PUT, T-Block и коммуникационные модули (CM)) не имеют функций обеспечения безопасности. Эти типы коммуникации должны быть защищены путем ограничения физического доступа. Если злоумышленник сможет получить физический доступ к сетям через эти типы коммуникации, то он сможет при необходимости читать и записывать данные.

Для получения информации и рекомендаций по обеспечению безопасности см. Operational Guidelines for Industrial Security (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) на сайте поддержки Siemens.

PROFINET

PROFINET используется для обмена данными через программу пользователя с другими участниками процесса коммуникации по сети Ethernet:

- В S7-1200 PROFINET поддерживает 16 IO-устройств с максимум 256 субмодулями, а PROFIBUS позволяет 3 независимых PROFIBUS DP ведущих устройства с поддержкой 32 ведомых на каждое ведущее устройство DP, с максимум 512 модулями на каждое ведущее устройство DP.
- коммуникация S7
- протокол дейтаграмм пользователя User Datagram Protocol (UDP)
- ISO on TCP (RFC 1006)
- протокол управления передачей данных Transport Control Protocol (TCP)

PROFINET IO контроллер

В качестве IO-контроллера, используя PROFINET IO, CPU выполняет обмен данными максимум с 16 PN устройствами в локальной PN сети или через PN/PN коммутатор (соединение). Дополнительную информацию можно найти по адресу PROFIBUS and PROFINET International, PI (www.us.profinet.com).

PROFIBUS

PROFIBUS используется для обмена данными через программу пользователя с другими участниками процесса коммуникации по сети PROFIBUS:

- С CM 1242-5 CPU работает в качестве ведомого устройства PROFIBUS DP.
- С CM 1243-5 CPU работает в качестве ведущего устройства PROFIBUS DP класса 1.
- Ведомые устройства PROFIBUS DP, ведущие устройства PROFIBUS DP, а также AS-i (3 коммуникационных модуля с левой стороны), и PROFINET являются отдельными коммуникационными сетями, которые не ограничивают друг друга.

ASi

S7-1200 CM 1243-2 ведущее устройство AS-i обеспечивает подключение S7-1200 CPU к сети AS-i.

Межпроцессорная связь через S7

Можно создать коммуникационное соединение со станцией партнера и использовать инструкции GET и PUT, чтобы обмениваться данными с S7 CPU.

Коммуникации TeleService

В TeleService через GPRS инженерная станция, на которой установлен STEP 7, обменивается данными по GSM-сети и Интернет со станцией SIMATIC S7-1200 с CP 1242-7. Соединение работает через сервер телеуправления, который служит посредником и подключен к Интернету.

IO-Link

Ведущее устройство S7-1200 SM 1278 4xIO-Link позволяет подключать устройства IO-Link к S7-1200 CPU.

11.1 Асинхронные коммуникационные соединения

Обзор коммуникационных служб

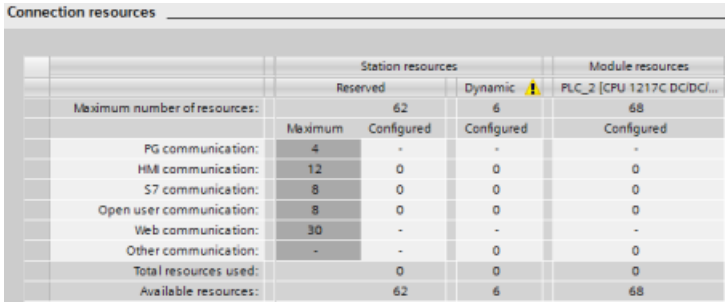
CPU поддерживает следующие коммуникационные службы:

Коммуникационная служба	Функциональность	Через PROFIBUS DP		Через Ethernet
		CM 1243-5 DP master module	CM 1242-5 DP slave module	
Коммуникация PG	Ввод в эксплуатацию, тестирование, диагностика	Да	Нет	Да
Коммуникация HMI	Оперативное управление и мониторинг	Да	Нет	Да
коммуникация S7	Обмен данными через сконфигурированные соединения	Да	Нет	Да
Маршрутизация функций PG	Например, тестирование и диагностика за границы сети	Нет	Нет	Нет
PROFIBUS DP	Обмен данными между ведущим и ведомым устройствами	Да	Да	Нет
PROFINET IO	Обмен данными между I/O контроллерами и I/O устройствами	Нет	Нет	Да
Веб-сервер	Диагностика	Нет	Нет	Да
SNMP [1] (Simple Network Management Protocol)	Стандартный протокол для диагностики сети и параметрирования	Нет	Нет	Да
S7-маршрутизация	Используя таблицы маршрутизации, участники процесса коммуникации могут связываться с любым устройством, даже если они находятся в разных подсетях S7.	Нет	Нет	Да
Открытая коммуникация по TCP/IP	Обмен данными через Industrial Ethernet по протоколу TCP/IP (с загружаемыми FB)	Нет	Нет	Да
Открытая коммуникация по ISO-on-TCP	Обмен данными через Industrial Ethernet по протоколу ISO-on-TCP (с загружаемыми FB)	Нет	Нет	Да
Открытая коммуникация по UDP	Обмен данными через Industrial Ethernet по протоколу UDP (с загружаемыми FB)	Нет	Нет	Да

[1] CPU поддерживает SNMP V1 без TRAP.

Доступные соединения

CPU поддерживает следующее максимальное количество одновременных, асинхронных коммуникационных соединений для PROFINET и PROFIBUS. Максимальное количество ресурсов подключения, выделенных каждой категории, является фиксированным. Эти значения не могут быть изменены. Однако можно использовать шесть свободно доступных соединений, чтобы увеличить количество соединений для любой категории в соответствии с требованиями приложения.



Connection resources

	Station resources			Module resources
	Reserved		Dynamic	PLC_2 [CPU 1217C DC/DC/...
Maximum number of resources:		62	6	68
	Maximum	Configured	Configured	Configured
PG communication:	4	-	-	-
HMI communication:	12	0	0	0
S7 communication:	8	0	0	0
Open user communication:	8	0	0	0
Web communication:	30	-	-	-
Other communication:	-	-	0	0
Total resources used:		0	0	0
Available resources:		62	6	68

Примечание

Добавление модулей CM/CP не увеличивает общего числа коммуникационных соединений S7-1200.

Примечание

Соединения OPC UA

Соединения OPC UA потребляют ресурсы шести "свободно доступных соединений". Соединения OPC UA отображаются в строке "Прочая коммуникация:" и в столбце "Динамическая". Убедиться, что для приложения имеется достаточно соединений.

На основании выделенных ресурсов подключения, для каждого устройства доступно следующее число соединений:

	Программатор (PG)	**Человеко-машинный интерфейс (HMI)**	**GET/PUT клиент/сервер**	**Открытые пользовательские коммуникации**	**Веб-браузер**
Максимальное количество ресурсов подключения	4 (гарантированная поддержка одного PG)	12 (гарантированная поддержка четырех устройств HMI)	8	8	30 (гарантированная поддержка трех веб-браузеров)

Пример: У CPU есть четыре доступных ресурса подключения PG. В зависимости от текущих функций PG, он может фактически использовать 1, 2, 3 или 4 из доступных ресурсов подключения. Всегда можно использовать один PG.

Следующий пример относится к числу устройств HMI, как показано на рисунке ниже. Устройства HMI имеют 12 доступных ресурсов подключения. В зависимости от типа или

модели устройства HMI, и используемых функций HMI, каждое устройство HMI может фактически использовать 1, 2, или 3 из своих доступных ресурсов подключения. В зависимости от числа используемых доступных ресурсов подключения, возможно одновременное использование более 4 устройств HMI. Однако всегда гарантируется по крайней мере 4 устройства HMI. Устройство HMI может использовать свои доступные ресурсы подключения (по 1 из 3 в общей сложности) для следующих функций:

- чтение
- запись
- аварийные сообщения и диагностика

Это лишь пример. Фактическое число соединений может отличаться в зависимости от типа и версии устройства HMI.

Пример	**HMI 1**	**HMI 2**	**HMI 3**	**HMI 4**	**HMI 5**	**Общее число доступных ресурсов подключения**
Используемые ресурсы подключения	2	2	2	3	3	12

Примечание

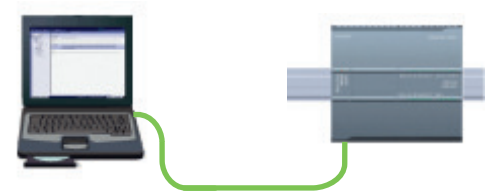
Подключения веб-сервера (HTTP): CPU предлагает соединения для нескольких веб-браузеров. Количество браузеров, которые может одновременно поддерживать CPU, зависит от того, сколько соединений запрашивает/использует один веб-браузер.

Примечание

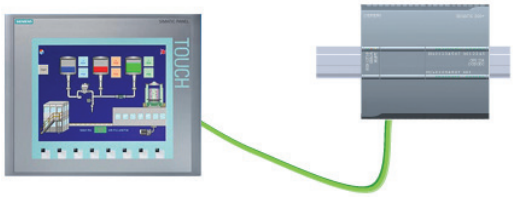
Коммуникационные подключения открытых коммуникационных соединений пользователя, S7-соединение, HMI, программатор и веб-сервер (HTTP) в зависимости от используемых функции могут занимать несколько ресурсов подключения.

11.2 PROFINET

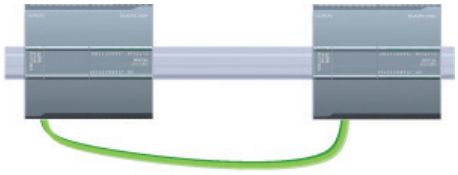
CPU может выполнять обмен с другими CPU, с программаторами, с HMI-устройствами и с устройствами, произведенными не фирмой Siemens, используя стандартные коммуникационные протоколы TCP.



Программатор, подключенный к CPU



Устройство HMI, подключенное к CPU



CPU, подключенный к другому CPU

Ethernet коммутатор

У CPU 1211C, 1212C и 1214C есть Ethernet порт и отсутствует встроенный Ethernet коммутатор. Прямое соединение между программатором или HMI и CPU не требует Ethernet коммутатора. Однако в сети больше чем с двумя CPU или устройствами HMI требуется Ethernet коммутатор.



① CPU 1215C

② Ethernet коммутатор CSM1277

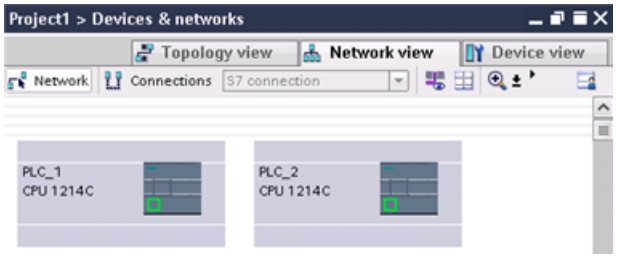
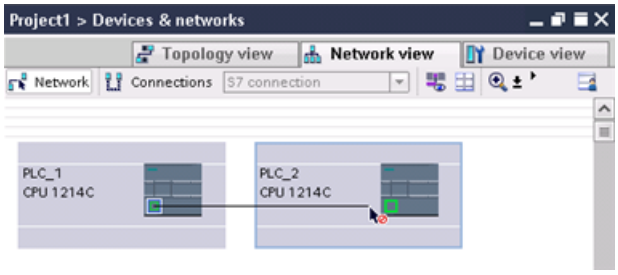
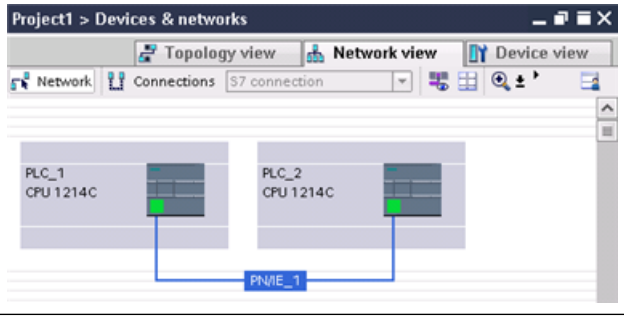
У CPU 1215C и 1217C есть встроенный Ethernet коммутатор с 2 портами. Может быть построена сеть с одним CPU 1215C и двумя другими CPU S7-1200. Также можно

использовать монтируемый в стойку Ethernet коммутатор CSM1277 с 4 портами для подключения нескольких CPU и устройств HMI.

11.2.1 Создание сетевого соединения

На вкладке просмотра сетевых соединений конфигурации устройства можно создавать сетевые соединения между устройствами. После создания сетевого соединения открыть вкладку "Свойства" окна инспектора для конфигурирования параметров сети.

Таблица 11-1 Создание сетевого соединения

Действие	Результат
Выбрать "Вид сети" для отображения устройств, которые должны быть соединены.	
Выбрать порт на одном устройстве и перетащить соединение на порт второго устройства.	
Отпустить кнопку мыши, чтобы создать сетевое соединение.	

11.2.2 Конфигурирование пути соединения между локальным и партнерским CPU

Соединение между локальным CPU и удаленным партнерским CPU определяет логическое назначение двух участников процесса коммуникации для организации коммуникационных служб. Соединение определяет следующие элементы:

- Задействованные участники процесса коммуникации (один активный, один пассивный)
- Тип соединения (напр., соединение с PLC, HMI или устройством)
- Путь соединения

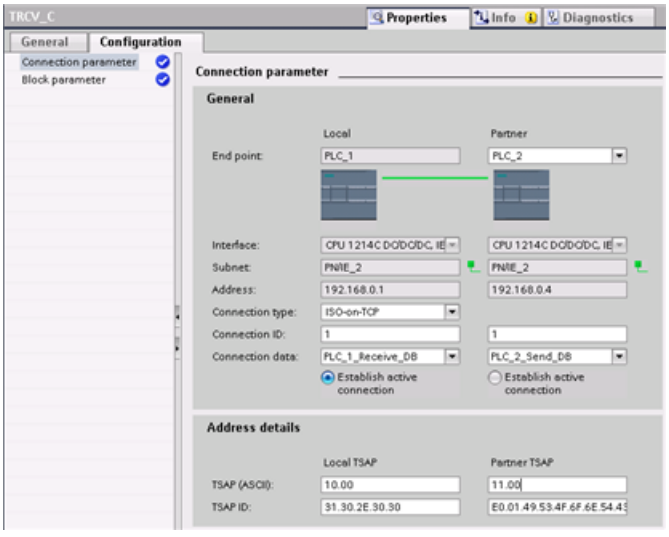
Участники процесса коммуникации выполняют инструкции, чтобы настроить и установить коммуникационное соединение. С помощью параметров задаются активная и пассивная конечные точки коммуникации. После настройки и установления соединения оно автоматически поддерживается и контролируется CPU.

Если соединение завершается (например, из-за разрыва линии), то активный партнер пытается восстановить его. Не требуется повторное выполнение коммуникационной инструкции.

Пути соединения

После вставки инструкции TSEND_C, TRCV_C или TCON в пользовательскую программу, окно инспектора выводит на экран свойства соединения каждый раз при выборе части инструкции. Параметры коммуникации определяются на вкладке "Конфигурация" в диалоге "Свойства" коммуникационной инструкции.

Таблица 11-2 Конфигурирование пути соединения (через свойства инструкции)

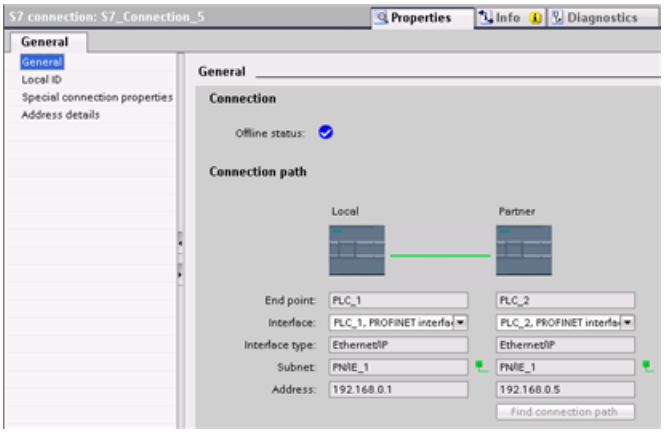
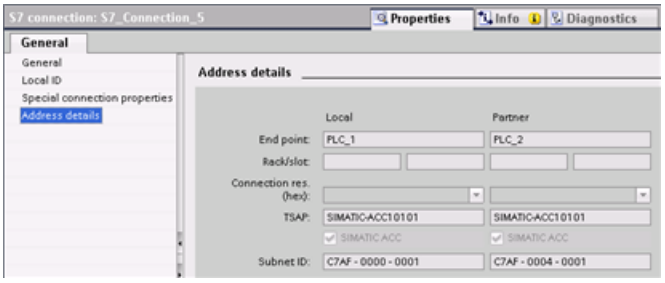
TCP, ISO-on-TCP и UDP	Свойства соединения
Для TCP, ISO-on-TCP и UDP Ethernet-протоколов, используют "Свойства" инструкции (TSEND_C, TRCV_C или TCON), чтобы сконфигурировать соединения локального и партнерского CPU. На рисунке показаны "Свойства соединения" на вкладке "Соединение" для ISO-on-TCP соединения.	

Примечание

При конфигурировании свойств соединения для CPU, STEP 7 позволяет либо выбрать определенный DB соединения в партнерском CPU (если такой существует), либо создать DB соединения для партнерского CPU. Партнерский CPU должен уже быть создан в проекте и не может быть "неопределенным" CPU.

Но в любом случае необходимо вставить инструкцию TSEND_C, TRCV_C или TCON в программу пользователя партнерского CPU. При вставке инструкции следует выбрать DB соединения, который был создан конфигурацией.

Таблица 11-3 Конфигурирование пути соединения для S7 коммуникаций (конфигурация устройства)

Коммуникация S7 (GET и PUT)	**Свойства соединения**
В случае коммуникации S7, для конфигурирования соединения между локальным и партнерским CPU используется редактор сети "Устройства и сети". Нажать кнопку "Выделено: Соединение", чтобы вызвать "Свойства". На вкладке "Общее" перечислено несколько свойств: • "Общее" (на изображении) • "Локальный ID" • "Специальные свойства соединения" • Адресная информация (на изображении)	 

В разделе "PROFINET" в пункте "Протоколы" (Страница 828) и в разделе "Коммуникация S7" " в пункте "Создание соединения S7" (Страница 1006) можно найти дополнительную информацию и список доступных коммуникационных инструкций.

Таблица 11-4 Параметры для соединения CPU

Параметр		**Определение**
Адрес		Назначенные IP-адреса
Общее	Конечная точка	Имя партнерского CPU (получатель)
	Интерфейс	Название интерфейсов
	Подсеть	Имя подсетей
	Тип интерфейса	*Только коммуникация S7*: Тип интерфейса
	Тип соединения	Тип протокола Ethernet
	Идентификатор соединения	ID номер
	Данные соединения	Память данных для локального CPU и партнерского CPU
	Установка активного соединения	Радиокнопка для выбора локального или партнерского ЦПУ в качестве активного соединения
Адресная информация	Конечная точка	*Только коммуникация S7*: Имя партнерского CPU (получатель)
	Монтажная стойка/слот	*Только коммуникация S7*: Адрес стойки и слота
	Ресурс подключения	*Только коммуникация S7*: Компонент TSAP для конфигурирования S7-соединение с S7-300 или S7-400 CPU
	Порт (десятич.):	TCP и UPD: Порт партнерского CPU в десятичном формате
	TSAP [1] и ID подсети:	ISO on TCP (RFC 1006) и S7-коммуникация: TSAP локального CPU и партнерского CPU в ASCII или шестн. формате

[1] Для конфигурирования соединения с S7-1200 CPU через ISO-on-TCP в расширении TSAP для пассивных участников процесса коммуникации следует использовать только символы ASCII.

Точки доступа службы уровня передачи (TSAP)

Использование TSAP позволяет протоколу ISO-on-TCP и S7-коммуникации использовать множественные соединения по единственному IP-адресу. TSAP однозначно идентифицируют эти соединения конечной точки связи с IP-адресом.

Используемые TSAP определяются в диалоговом окне "Параметры соединения" в пункте "Адресная информация". TSAP соединения в CPU вводится в поле "Локальный TSAP". TSAP для соединения в партнерском CPU вводится в поле "Партнерская TSAP".

Номера портов

Для протоколов TCP и UDP, в конфигурации параметров соединения локального (активного) CPU должен быть указан распределенный IP-адрес и номер порта (пассивного) партнерского CPU.

Используемые порты определяются в диалоговом окне "Параметры соединения" в пункте "Адресная информация". Порт соединения в CPU вводится в поле "Локальный порт". Порт для соединения в партнерском CPU вводится в поле "Партнерский порт".

11.2.3 Назначение IP-адресов

11.2.3.1 Назначение IP-адресов программатору и сетевым устройствам

Если программатор использует встроенную карту адаптера, подключенную к LAN предприятия, то программатор и CPU должны находиться в одной подсети. Подсеть назначается как комбинация из IP-адреса и маски подсети для устройства. Для получения помощи можно обратиться к сетевом администратору на месте.

ID сети это первые три октета IP-адреса, напр., **211.154.184**.16. Они определяют конкретную используемую IP-сеть. Маска подсети обычно имеет значение **255.255.255.0**. Но так как компьютер находится в локальной сети предприятия, маска подсети может принимать другие значения (напр., **255.255.254.0**) для установки однозначных подсетей. Маска подсети определяет через логическую операцию И с IP-адресом устройства границы IP-подсети.

Примечание

В Интернете, где программаторы, сетевые устройства и IP-маршрутизаторы взаимодействуют с устройствами по всему миру, должны быть назначены уникальные IP-адреса, чтобы избежать конфликта с другими пользователями сети. Следует связаться с ИТ-персоналом предприятия, который знаком с его сетями, для назначения ваших IP-адресов.

⚠ ПРЕДУПРЕЖДЕНИЕ

Несанкционированный доступ к CPU через веб-сервер

Несанкционированный доступ к CPU или установка неправильных значений для переменных PLC могут нарушить ход процесса и привести к тяжелым телесным повреждениям и/или материальному ущербу.

Поскольку активация веб-сервера позволяет авторизованным пользователям выполнять изменения рабочего состояния, записывать данные PLC и обновлять прошивку, Siemens рекомендует придерживаться следующих мер безопасности:

- Активировать доступ к веб-серверу только по протоколу HTTPS.
- Защитить сложным паролем идентификаторы пользователей веб-сервера (Страница 1028). Сложные пароли - это, по крайней мере, десять символов, представляющих комбинацию букв, цифр и специальных символов, не являющиеся словами, которые могут быть найдены в словаре, и не являющиеся именами или идентификаторами, которые могут быть получены из персональных данных. Следует хранить пароль в секрете и часто его изменять.
- Не расширять предустановленные минимальные права для пользователя "Все".
- Выполнить поиск ошибок и проверку диапазонов для переменных в программной логике, так как пользователи веб-страниц могут устанавливать для переменных PLC недопустимые значения.

Примечание

Если CPU подключается не к локальной сети предприятия, то рекомендуется использовать вторую карту для доступа в сеть. Такая конфигурация в первую очередь рекомендуется для начального тестирования или для контроля на этапе ввода в эксплуатацию.

Назначение или проверка IP-адреса программатора с помощью функции "Сетевое окружение" (на рабочем столе)

Выполнить следующие действия для назначения или проверка IP-адреса программатора:

1. Открыть панель управления через меню "Пуск".
2. Открыть „Центр управления сетями и общим доступом“" и выбрать "Подключение по локальной сети" для подключенного к CPU сетевого адаптера
3. Нажать кнопку "Свойства" в диалоговом окне "Состояние подключения по локальной сети".
4. Выбрать в диалоговом окне "Подключение по локальной сети - свойства" в поле "Отмеченные компоненты используются этим соединением:" элемент "Протокол интернета версии 4 (TCP/IPv4)".
5. Нажать кнопку "Свойства".
6. Выбрать вариант "Получить IP-адрес автоматически" или "Использовать следующий "IP-адрес", чтобы ввести статический IP-адрес.
7. Если был выбран вариант "Использовать следующий "IP-адрес", то необходимо установить IP-адрес и маску подсети:
   - IP-адрес должен быть выбран таким образом, чтобы использовались те же ID сети и маска подсети, что и для CPU. Пример: Если die IP-адрес CPU **192.168.0**.1, то в качестве IP-адреса можно установить **192.168.0**.200.
   - Выбрать маску подсети **255.255.255**.0.
   - Поле "Основной шлюз" оставить пустым.

Теперь может быть установлена связь с CPU.

Примечание

Сетевая интерфейсная карта и CPU должны находиться в одной подсети, чтобы STEP 7 смогла найти CPU и установить с ним связь.

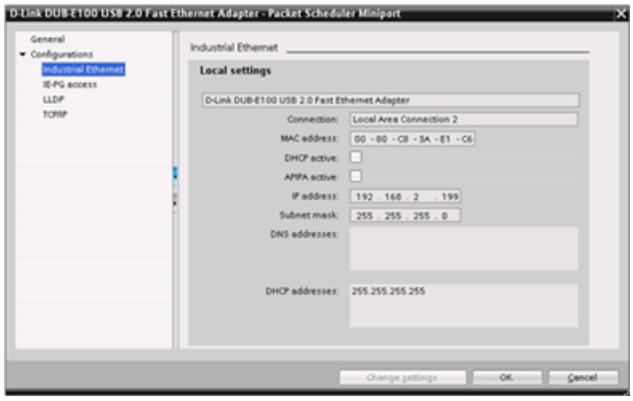
Обратиться к ИТ-персоналу предприятия для получения помощи при настройке конфигурации сети, позволяющей подключаться к S7-1200 CPU.

11.2.3.2 Проверка IP-адреса программатора

Для проверки MAC- и IP-адреса программатора используются следующие команды меню:

1. Раскрыть в дереве узел "Доступ в режиме онлайн"
2. Кликнуть правой кнопкой мыши по требуемой сети и выбрать "Свойства".
3. Раскрыть в диалоговом окне "Сеть" пункт "Конфигурации" в выбрать "Industrial Ethernet".

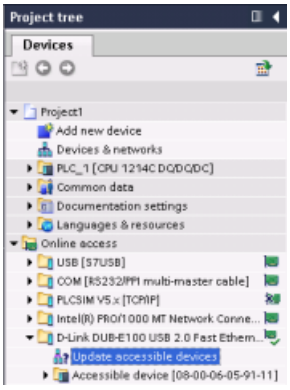
Здесь отображаются MAC- и IP-адреса программатора.



11.2.3.3 Назначение IP-адресов для CPU в режиме онлайн

Можно назначить IP-адрес сетевому устройству в режиме онлайн. Это особенно полезно при первом конфигурировании устройства.

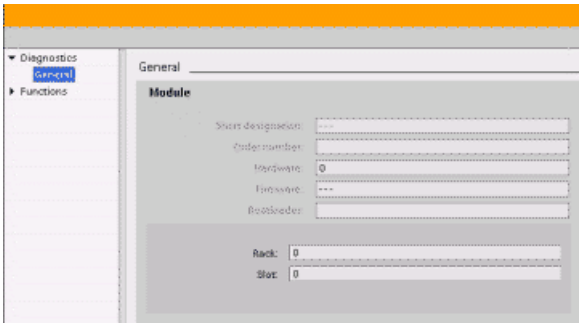
1. Проверить в дереве проекта, нет ли у CPU сконфигурированного IP-адреса. Раскрыть узел "Доступ к режиме онлайн" > <Адаптерная плата для сети, в которой находится устройство>, и выполнит двойной клик по "Обновить доступные устройства".



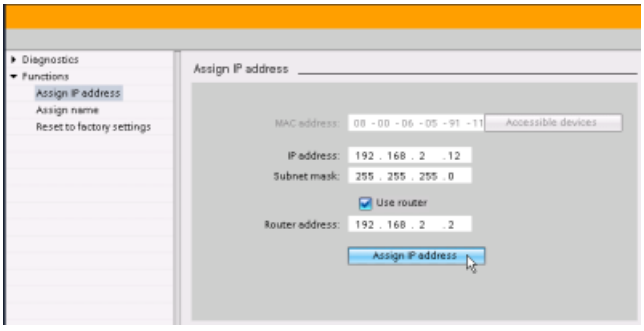
Если STEP 7 показывает MAC-адрес вместо IP-адреса, значит IP-адрес не был назначен.

2. Дважды кликнуть под требуемым доступным устройством по "Онлайн и диагностика".

3. Выбрать в диалоговом окне "Онлайн и диагностика" опцию "Функции" > "Назначение IP-адреса".

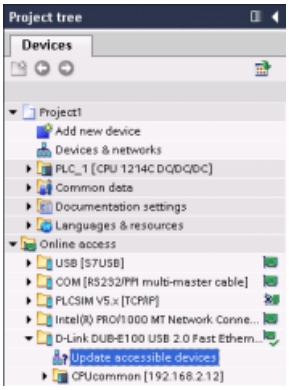


4. Ввести в поле "IP-адрес" новый IP-адрес и нажать кнопку "Назначить IP-адрес".



5. Проверить в дереве проекта, присвоила ли STEP 7 новый IP-адрес для CPU.

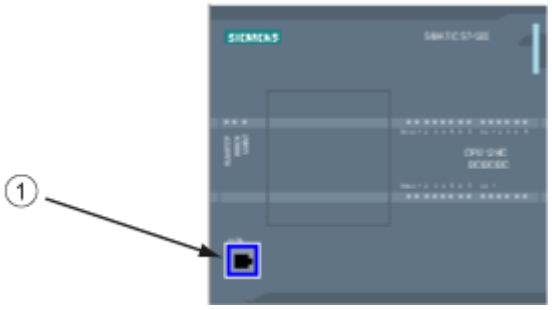
Двойной клик по "Обновить доступные устройства", чтобы показать сконфигурированный ранее IP-адрес.



11.2.3.4 Конфигурирование IP-адреса для CPU в проекте

Конфигурирование интерфейса PROFINET

Чтобы сконфигурировать параметры для интерфейса PROFINET, выбрать зеленое поле PROFINET на CPU. В окне инспектора отображается вкладка "Свойства" для порта PROFINET.



① PROFINET порт

Конфигурирование IP-адреса

Ethernet адрес (MAC-адрес): В сети PROFINET каждому устройству производителем для идентификации назначается MAC-адрес (Media Access Control address). MAC-адрес состоит из шести групп по две шестнадцатеричных цифры в каждой, разделенных дефисами (-) или двоеточиями (:) в порядке передачи (например, 01-23-45-67-89-AB или 01:23:45:67:89:AB).

IP-адрес: Каждое устройство должно также иметь адрес протокола Интернет (Internet Protocol, IP). Этот адрес позволяет устройству передавать данные по более сложной, маршрутизированной сети.

Каждый IP-адрес разделен на четыре 8-разрядных сегмента и представлен в десятичном формате с разделением точкой (например, 211.154.184.16). Первая часть IP-адреса используется для ID сети (идентификатор активной сети), а вторая часть адреса - это идентификатор хоста (уникальный для каждого устройства в сети). IP-адрес 192.168.x.y является стандартным обозначением, распознаваемым как часть частной сети, которая не находится в Интернете.

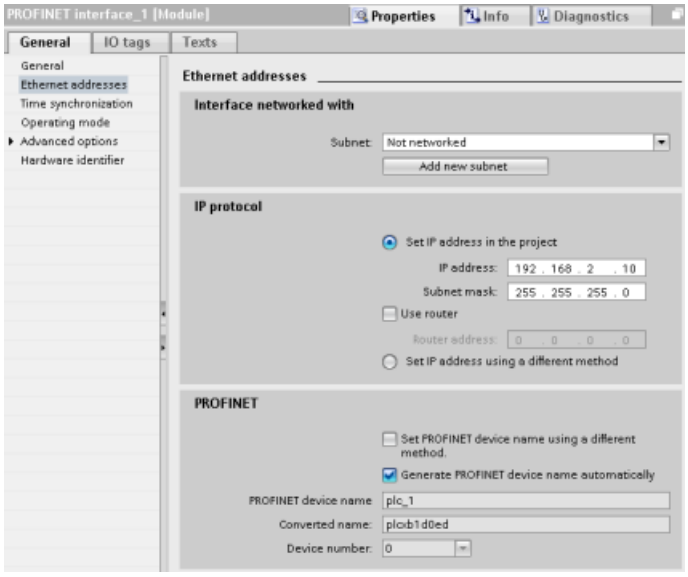
Маска подсети: Подсеть является логическим группированием соединенных между собой сетевых устройств. Узлы подсети обычно расположены в непосредственно близости друг от друга в локальной сети (LAN). Маска (известная как маска подсети или сетевая маска) определяет границы IP подсети.

Маска подсети 255.255.255.0 обычно подходит для небольшой локальной сети. Это означает, что у всех IP-адресов в этой сети должны быть одинаковые первые 3 октета. Различные устройства в этой сети идентифицируются по последнему октету (8-разрядное поле). Пример является назначение маски подсети 255.255.255.0 и IP-адреса от 192.168.2.0 до 192.168.2.255 для устройств небольшой локальной сети.

Соединение между различными подсетями осуществляется только через маршрутизатор. Если используются подсети, то должен использоваться IP-маршрутизатор.

IP-маршрутизатор: Маршрутизаторы представляют собой канал связи между LAN. Используя маршрутизатор, компьютер в LAN может отправлять сообщения в другие сети, которые в свою очередь относятся к другим LAN. Если место назначения данных не в LAN, то маршрутизатор передает данные другой сети или группе сетей, где они могут быть доставлены в место назначения.

Для передачи и приема пакетов данных, маршрутизаторы используют IP-адреса.



Свойства IP-адресов: Выбрать в диалоговом окне "Свойства" элемент "Ethernet-адреса". STEP 7 отображает диалог для конфигурирования Ethernet адреса, в котором программному проекту присваивается IP-адрес модуля CPU, в который загружается проект.

Таблица 11-5 Параметры для IP-адреса

Параметр		**Описание**
Подсеть	Имя подсети, к которой подключено устройство. Нажать кнопку "Добавить новую подсеть", чтобы создать новую подсеть. Установкой по умолчанию является "Не соединено". Возможны два типа соединений: • Установка по умолчанию "Не соединено" предоставляет локальное соединение. • Подсеть необходима, если сеть содержит два или более устройств.	
Протокол IP	IP-адрес	Назначенный IP-адрес CPU
	Маска подсети	Назначенная маска подсети
	Использовать маршрутизатор IP	Установить флажок, если используется IP-маршрутизатор
	Адрес маршрутизатора	Назначенный IP-адрес для маршрутизатора, если он имеется

Примечание

Все IP-адреса конфигурируются при загрузке проекта. Если у CPU нет предварительно сконфигурированного IP-адреса, следует связать проект с MAC-адресом целевого устройства. Если CPU подключен к маршрутизатору или сети, то следует ввести и IP-адрес маршрутизатора.

Радиокнопка "Установить IP-адрес другими способом" позволяет изменять IP-адрес онлайн или при помощи инструкции "T_CONFIG (Страница 919)" после того, как программа будет загружена в CPU. Этот метод назначения IP-адреса действует только для CPU.

⚠ ПРЕДУПРЕЖДЕНИЕ

Загрузка аппаратной конфигурации с "Установить IP-адрес другим способом"

После загрузки аппаратной конфигурации с включенной опцией "Установить IP-адрес другим способом", невозможно перевести рабочий режим CPU из RUN в STOP или из STOP в RUN.

Пользовательское оборудование продолжает работать при этих условиях, что может привести к неожиданной реакции машины или процесса, которая в свою очередь может стать причиной смерти, тяжких телесных повреждений или материального ущерба, если не будут предприняты надлежащие меры предосторожности.

Следует убедиться, что IP-адрес(-а) устройств CPU установлены, перед использованием CPU в реальной среде автоматизации. Это может быть сделано при помощи пакета программирования STEP 7, SIMATIC Automation Tool или подключенного устройства HMI в сочетании с инструкцией T_CONFIG.

⚠ ПРЕДУПРЕЖДЕНИЕ

Условия, при которых возможна остановка сети PROFINET

При изменении IP-адреса CPU онлайн или из пользовательской программы, возможно состояние, в котором сеть PROFINET будет остановлена.

Если IP-адрес CPU будет изменен на IP-адрес, находящийся вне подсети, то сеть PROFINET потеряет возможность выполнять коммуникацию, и весь обмен данными остановится. Устройства пользователя могут быть настроены таким образом, что они продолжат работать при таких условиях. Сбой коммуникации PROFINET может привести к неожиданной реакции машины или процесса, которая в свою очередь может стать причиной смерти, тяжких телесных повреждений или материального ущерба, если не будут предприняты надлежащие меры предосторожности.

Если IP-адрес должен быть изменен вручную, следует убедиться, что новый IP-адрес находится в подсети.

Конфигурирование порта PROFINET

По умолчанию CPU конфигурирует порты интерфейса PROFINET для автоопределения. Для правильной работы автоопределения, обе станции должны быть настроены для автоопределения. Если одна станция имеет постоянную конфигурацию (напр., дуплексный режим с 100 Мбит/с), а другая станция настроена для автоопределения, автоопределение не будет работать, что приведет к полудуплексному режиму.

Для снятия этого ограничения автоопределения, S7-1200 предлагает опцию для отключения автоопределения. После отключения автоопределения, S7-1200 автоматически переходит на дуплексный режим при 100 Мбит/с.

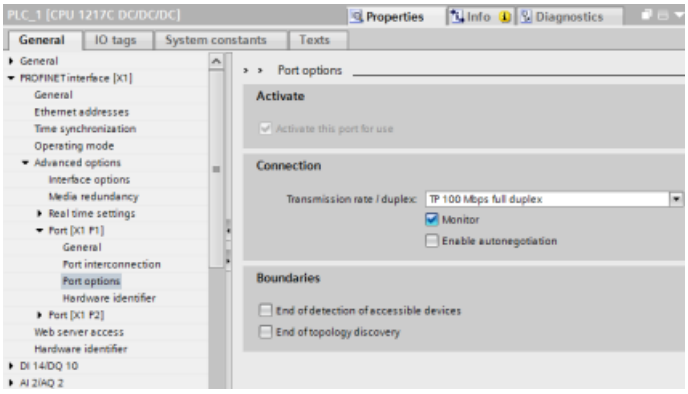
Скорость передачи и дуплекс могут быть установлены на постоянное значение для каждого порта:

1. Выбрать расширенные опции и порт для конфигурирования. После выбрать "Опции порта".
2. Выбрать в пункте "Соединение" в поле "Скорость передачи данных/дуплекс" одну из следующих опций:
   - Автоматически: CPU и партнерское устройство используют автоопределение для получения скорости передачи и дуплекса порта.
   - TP 100 Мбит/с полный дуплекс: Если автоопределение отключается, то порт работает в дуплексном режиме с 100 Мбит/с. Если включить автоопределение, то порт может работать с дуплексном режиме с 100 Мбит/с или другой скоростью передачи данных/дуплексом, которые выбираются между CPU и партнерским устройством через автоопределение (ссли выбирается "Мониторинг", то в буфер диагностики заносится сообщение (см. ниже)).

3. Мониторинг: Если установить этот флажок, то в буфер диагностики заносится сообщение при возникновении на порту одной из следующих ситуаций:
   - Соединение на порту не может быть установлено
   - Отказ установленного соединения
   - Выбирается "TP 100 Мбит/с полный дуплекс" как скорость передачи/дуплекс, а CPU через автоопределение устанавливает соединение с согласованной скоростью передачи, отличной от 100 Мбит/с или с дуплексом, равным полудуплексу.
4. Активация автоопределения: Если в поле "Скорость передачи данных/дуплекс" выбирается опция для полного дуплекса при 100 Мбит/с, то автоопределение может быть отключено. Для выключения автоопределения, снять флажок "Активировать автоопределение".

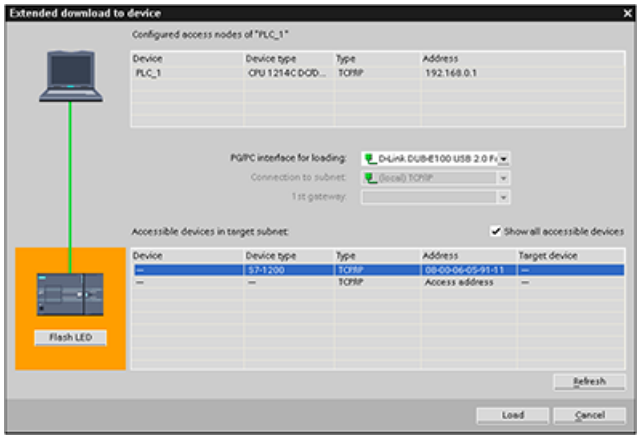
Примечание

Если автоопределение не будет отключено, то CPU и партнерское устройство используют автоопределение для получения скорости передачи и дуплекса порта.



11.2.4 Тестирование сети PROFINET

После завершения конфигурирования следует загрузить программу (Страница 209) в CPU. Все IP-адреса конфигурируются при загрузке проекта.



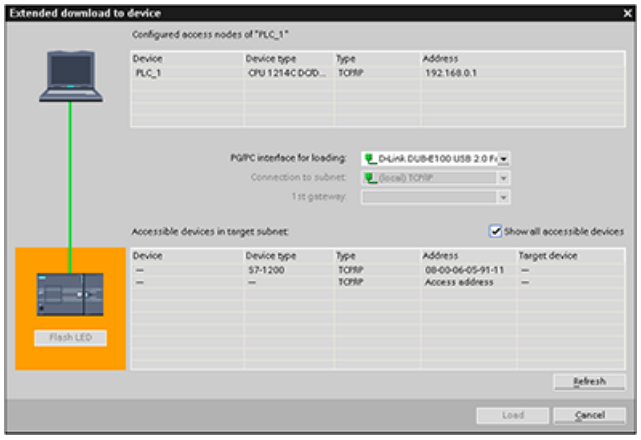
Назначение IP-адреса устройству онлайн

У S7-1200 CPU нет предварительно сконфигурированного IP-адреса. Поэтому необходимо вручную назначить IP-адрес для CPU.

- Как назначить устройству в режиме онлайн IP-адрес, можно узнать в разделе "Конфигурация устройства: назначение IP-адреса CPU в режиме онлайн" (Страница 817).
- Чтобы назначить IP-адрес в проекте, необходимо задать IP-адрес в конфигурации устройства, сохранить конфигурацию и загрузить ее в целевую систему. Дополнительную информацию см. в Конфигурация устройства: конфигурирование IP-адреса для CPU в проекте" (Страница 819).

Использование диалога "Расширенная загрузка в устройство" для запроса о подключенных сетевых устройствах

Функция CPU S7-1200 "Загрузка в устройство" и ее диалоговое окно "Расширенная загрузка в устройство" могут показать все доступные сетевые устройства, а также всем ли устройствам назначены уникальные IP-адреса. Для отображения всех доступных и имеющихся устройств с назначенными им MAC- и IP-адресами, установить флажок "Показать все доступные устройства".



Если требуемого сетевого устройства нет в этом списке, то связь с этим устройством была по какой-то причине прервана. Это устройство и сеть должны быть обследованы на предмет наличия ошибок в оборудовании или в конфигурации.

11.2.5 Определение Ethernet-адреса (MAC-Adresse) у CPU

В сетях PROFINET адрес на уровне управления доступом к среде (MAC-адрес) является идентификатором, назначаемым сетевому интерфейсу производителем для однозначной идентификации. MAC-адрес обычно содержит зарегистрированный идентификационный номер производителя.

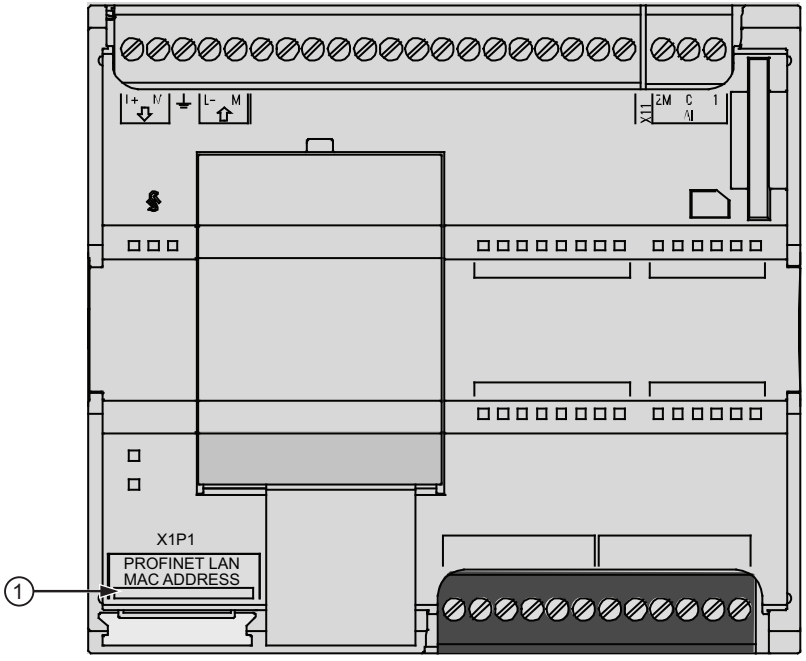
Стандартный (IEEE 802.3) формат для печати MAC-адресов открытым текстом представляет собой шесть групп из двух шестнадцатеричных цифр в каждой,

разделенных дефисами (-) или двоеточиями (:), в порядке передачи, (например, 01-23-45-67-89-ab или 01:23:45:67:89:ab).

Примечание

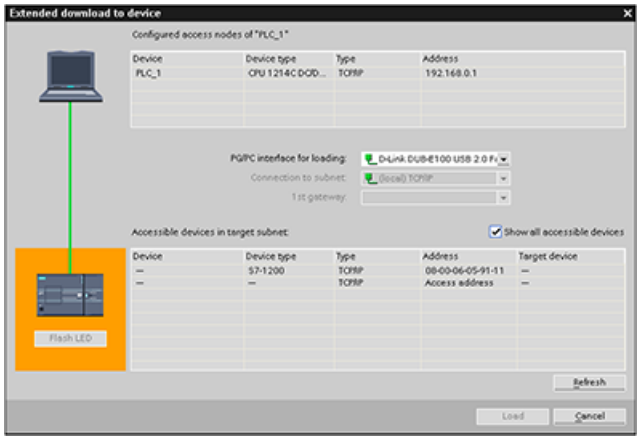
Для каждого CPU на заводе устанавливается постоянный, уникальный MAC-адрес. MAC-адрес CPU не может быть изменен.

MAC-адрес указан на лицевой стороне в нижнем левом углу CPU. Следует открыть нижнюю крышку, чтобы можно было прочесть MAC-адрес.



① MAC-адрес

Изначально у CPU нет IP-адреса, а только установленный на заводе MAC-адрес. Для PROFINET коммуникации у всех устройств должен быть уникальный IP-адрес.



Функция CPU "Загрузка в устройство" и ее диалоговое окно "Расширенная загрузка" могут показать все доступные сетевые устройства, а также всем ли устройствам были назначены уникальные IP-адреса. Это диалоговое окно отображает все доступные и имеющиеся устройства с назначенными им MAC- и IP-адресами. MAC-адреса особенно важны для идентификации устройств, у которых отсутствует требуемый уникальный IP-адрес.

11.2.6 Конфигурирование NTP синхронизации (Network Time Protocol, NTP)

⚠ ПРЕДУПРЕЖДЕНИЕ

Опасность несанкционированного доступа к сетям пользователя при синхронизации по протоколу NTP (Network Time Protocol)

Если злоумышленник получит доступ к сетям пользователя посредством синхронизации по протоколу сетевого времени (NTP), то он может начать управлять процессом, изменив системное время CPU. Такая передача контроля над процессом может привести к смерти, серьезным травмам или материальному ущербу.

По умолчанию S7-1200 CPU отключает функцию NTP-клиента. Если включить функцию NTP, то только установленные пользователем IP-адреса могут использоваться в качестве NTP-сервера. Функция NTP должна быть настроена таким образом, чтобы системное время CPU могло бы корректироваться с удаленных серверов.

S7-1200 CPU поддерживает прерывания и операции по времени, которые зависят от точности системного времени CPU. Если настраивается NTP и разрешается синхронизацию времени с сервера, то следует убедиться, что сервер является доверенным источником. Невыполнение этого требования может привести к нарушениям в системе безопасности, которые могут позволить неизвестному пользователю взять на себя управление процессом, переставив системное время CPU.

Для получения информации и рекомендаций по обеспечению безопасности см. Operational Guidelines for Industrial Security (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) на сайте поддержки Siemens.

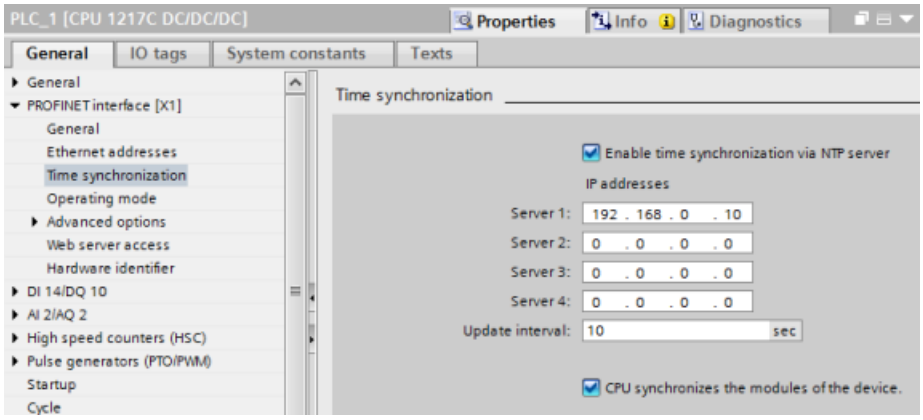
Протокол сетевого времени (NTP) широко используется, чтобы синхронизировать часы компьютерных систем с Интернет-серверами времени. При использовании метода NTP, CPU запрашивает время через регулярные интервалы (в клиентском режиме) у NTP-сервера в подсети (LAN). На основании ответов серверов определяется самое надежное и точное время и производится его синхронизация со станцией.

Преимущество этого метода заключается в том, что он позволяет синхронизировать время между подсетями.

Можно настроить IP-адреса до четырех серверов NTP. Интервал обновления определяет интервал между запросами времени (в секундах). Значение интервала может выбираться в диапазоне от 10 секунд до одного дня.

При использовании метода NTP обычно передается UTC (Universal Time Coordinated); это соответствует GMT (Greenwich Mean Time).

В окне свойств конфигурации устройства CPU выбрать элемент "Синхронизация времени". STEP 7 отображает диалоговое окно "Синхронизация времени":



Примечание

CPU получает все IP-адреса при загрузке проекта.

Таблица 11-6 Параметры для синхронизации времени

Параметр	Определение
Активировать синхронизацию времени через NTP-сервер	Установить флажок для включения синхронизации времени через NTP-сервер.
Сервер 1	Назначенный IP-адрес для сетевого сервера времени 1
Сервер 2	Назначенный IP-адрес для сетевого сервера времени 2
Сервер 3	Назначенный IP-адрес для сетевого сервера времени 3
Сервер 4	Назначенный IP-адрес для сетевого сервера времени 4
Интервал обновления для синхронизации времени	Значение интервала (в секундах)
CPU синхронизирует модули устройства.	Установить флажок для синхронизации часов CP с часами CPU.

11.2.7 Пусковой период, присвоение имени и назначение адреса PROFINET устройствам

PROFINET IO может увеличить пусковой период для системы (конфигурируемый тайм-аут). Большее количество устройств и медленные устройства влияют на время, которое необходимо, чтобы переключиться в RUN.

В V4.0 и более поздних версиях может быть максимум 16 PROFINET IO устройств в S7-1200 PROFINET сети.

Каждая станция (или IO-устройство) запускается независимо, и это влияет на общий пусковой период CPU. Если конфигурируемый тайм-аут устанавливается слишком маленьким, то общий пусковой период CPU может оказаться недостаточным для всех станций, чтобы завершить процесс запуска. Если происходит подобная ситуация, то возникают ложные ошибки станций.

"Время назначения параметров для распределенных I/O"можно найти в свойствах CPU в разделе "Пуск". Значение по умолчанию для конфигурируемого тайм-аута равно 60 000 мс (1 минута). Это время может конфигурироваться пользователем.

Имена и адреса устройств PROFINET в STEP 7

Все устройства PROFINET **должны** иметь имя устройства и IP-адрес. Имя устройства и IP-адреса устанавливаются в STEP 7. Имена устройств загружаются в IO-устройства, используя PROFINET DCP (Discovery and Configuration Protocol, протокол обнаружения и конфигурирования).

Назначение адресов PROFINET при запуске системы

Контроллер передает имена устройства в сеть, устройства реагируют через свои MAC-адреса. После контроллер назначает устройству через PROFINET DCP протокол IP-адрес:

- Если у MAC-адреса есть сконфигурированный IP-адрес, то станция выполняет запуск.
- Если у MAC-адреса нет сконфигурированного IP-адреса, STEP 7 назначает адрес, который сконфигурирован в проекте, и станция затем выполняет запуск.
- Если существует проблема с этим процессом, то возникает ошибка станции, и запуск не выполняется. Эта ситуация приводит к превышению сконфигурированного значения тайм-аута.

11.2.8 Открытые коммуникационные соединения пользователя

11.2.8.1 Протоколы

Встроенный PROFINET порт CPU поддерживает несколько коммуникационных стандартов по Ethernet сети:

- протокол управления передачей данных Transport Control Protocol (TCP)
- ISO on TCP (RFC 1006)
- протокол дейтаграмм пользователя User Datagram Protocol (UDP)

Таблица 11-7 Протоколы и коммуникационные инструкции

Протокол	Примеры использования	Запись данных в область получения	Коммуникационные инструкции	Тип адресации
TCP	Межпроцессорная коммуникация Передача телеграмм	Ad hoc режим	Только TRCV_C иTRCV	Назначает номера портов локальным (активным) и партнерским (пассивным) устройствам
		Получение данных с указанной длиной	TSEND_C, TRCV_C, TCON, TDISCON, TSEND и TRCV	

Протокол	Примеры использования	Запись данных в область получения	Коммуникационные инструкции	Тип адресации
ISO on TCP	Межпроцессорная коммуникация Фрагментация и повторная сборка сообщений	Ad hoc режим	Только TRCV_C иTRCV	Назначает TSAP локальным (активным) и партнерским (пассивным) устройствам
		Управляется протоколом	TSEND_C, TRCV_C, TCON, TDISCON, TSEND и TRCV	
UDP	Межпроцессорная коммуникация Коммуникация в пользовательской программе	Протокол дейтаграмм пользователя (User Datagram Protocol, UDP)	TUSEND и TURCV	Назначает номера портов локальным (активным) и партнерским (пассивным) устройствам, но речь не идет о выделенном соединении
Коммуникация S7	Межпроцессорная коммуникация Чтение данных из CPU/запись в CPU	Передача и получение данных с указанной длиной	GET и PUT	Назначает TSAP локальным (активным) и партнерским (пассивным) устройствам
PROFINET IO	Коммуникация между CPU и PROFINET IO-устройством	Передача и получение данных с указанной длиной	Встроенные	Встроенный

См. также

Новые функции (Страница 35)

11.2.8.2 TCP и ISO on TCP

Протокол управления передачей данных (Transport Control Protocol, TCP) - это стандартный протокол, описанный RFC 793: Transmission Control Protocol. Основное назначение TCP - обеспечить надежное и безопасное соединение между парами процессов. Этот протокол обладает следующими особенностями:

- Эффективный коммуникационный протокол, так как он тесно связан с оборудованием
- Подходит для данных среднего и большого объема (до 8192 байтов)
- Предоставляет значительно больше средств для приложений, в особенности, восстановление после ошибки, управления потоком и надежность
- Ориентированный на соединение протокол
- Может использоваться очень гибко со сторонними системами, которые поддерживают только TCP
- Поддерживает маршрутизацию
- Применимы только статические длины данных.
- Сообщения квитируются.
- Адресация приложений осуществляется через номера портов.

- Большинство протоколов приложений пользователя, например, TELNET и FTP, используют TCP.
- Для управления данными используется программирование из-за интерфейса программирования SEND / RECEIVE.

Международная организация по стандартизации (ISO) на Протоколе управления передачей (TCP) (RFC 1006) (ISO on TCP) является механизмом, который позволяет ISO приложениям быть перенаправленным в TCP/IP сеть. Этот протокол обладает следующими особенностями:

- Тесно связанный с оборудованием, эффективный коммуникационный протокол
- Подходит для данных среднего и большого объема (до 8192 байтов)
- В отличие от TCP, сообщения обладают идентификацией завершения данных и ориентированы на сообщение.
- Поддерживает маршрутизацию, возможно использование в WAN
- Возможны динамические длины данных.
- Для управления данными используется программирование из-за интерфейса программирования SEND / RECEIVE.

Использование TSAP позволяет протоколу TCP использовать множественные соединения по единственному IP-адресу (до 64 К соединений). С RFC 1006 TSAP однозначно идентифицируют эти соединения конечной точки связи с IP-адресом.

11.2.8.3 Коммуникационные службы и используемые номера портов

S7-1200 CPU поддерживает протоколы, перечисленные в таблице ниже. Для каждого протокола CPU назначает адресные параметры, соответствующий сеансовый уровень, а также коммуникационную роль и направление коммуникаций.

Эта информация позволяет выбрать меры обеспечения безопасности для защиты системы автоматизации согласно используемым протоколам (например, брандмауэр). Только Ethernet- и PROFINET-сети поддерживают меры обеспечения безопасности. Так как для PROFIBUS отсутствуют меры обеспечения безопасности, таблица не содержит PROFIBUS-протоколы.

Таблица ниже показывает различные уровни и протоколы, которые использует CPU:

Протокол	**Помер порта**	**(2) Канальный уровень (4) Транспорт-ный уровень**	**Функция**	**Описание**
PROFINET-протоколы				
DCP (Discovery and Configuration Protocol, протокол обнаружения и конфигурирования)	Не применимо	(2) Ethernet II и IEEE 802.1Q и Ethertype 0x8892 (PROFINET)	Обнаружение и конфигурирование для доступных устройств PROFINET	PROFINET использует DCP, чтобы обнаружить устройства и предложить основные настройки. DCP использует специальный групповой MAC-адрес: xx-xx-xx-01-0E-CF, xx-xx-xx = Organizationally Unique Indentifier (идентификатор производителя)
LLDP (Link Layer Discovery Protocol, протокол обнаружения канального уровня)	Не применимо	(2) Ethernet II и IEEE 802.1Q и Ethertype 0x88CC (PROFINET)	PROFINET Link Layer Discovery Protocol	PROFINET использует протокол LLDP для обнаружения и управления взаимодействием между расположенными рядом друг с другом PROFINET устройствами. LLDP использует специальный групповой MAC-адрес: 01-80-C2-00-00-0E

11.2.8.4 Ad hoc режим

Как правило, протоколы TCP и ISO-on-TCP принимают пакеты данных фиксированной длины от 1 до 8192 байт. Однако коммуникационные инструкции TRCV_C и TRCV также поддерживают коммуникационный Ad hoc режим, который позволяет принимать пакеты данных переменной длины от 1 до 1472 байтов.

Примечание

Если данные хранятся в "оптимизированном" DB (только символьная адресация), то данные могут быть приняты только в массивах данных типа Byte, Char, USInt иSInt.

Для конфигурирования инструкции TRCV_C или TRCV для Ad hoc режима, установить входной параметр ADHOC инструкции.

Если инструкция TRCV_C или TRCV вызывается в Ad hoc режиме редко, то за один вызов можно получить несколько пакетов. Пример: Если необходимо получить пять 100-байтовых пакетов за один вызов, то TCP доставит эти пять пакетов как один 500-байтовый пакет, в то время как ISO-on-TCP разобьет пакеты на пять 100-байтовых пакетов.

11.2.8.5 Идентификаторы (ID) для открытых коммуникационных соединений пользователя

При вставке PROFINET инструкции TSEND_C, TRCV_C или TCON в программу пользователя, STEP 7 создает DB экземпляра, чтобы сконфигурировать коммуникационный канал (или соединение) между устройствами. Параметры соединения настраиваются в "Свойствах" (Страница 812) инструкции. Среди параметров есть ID для этого соединения.

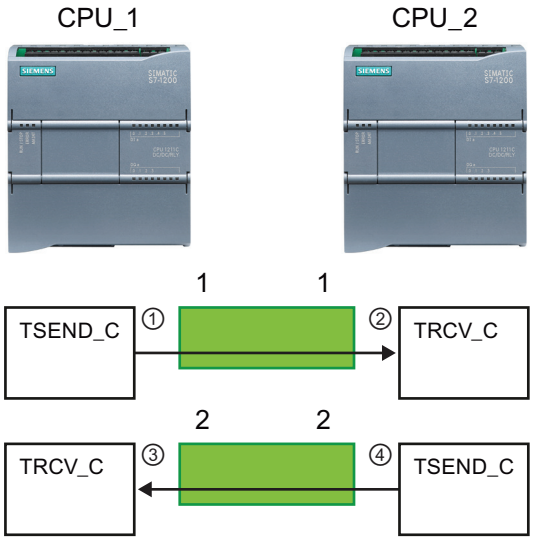
- ID соединения должен быть уникальным для CPU. У каждого создаваемого соединения должны быть отличающиеся от других DB и ID соединения.
- И локальный CPU, и партнерский CPU могут использовать один и тот же идентификационный номер соединения для одного и того же соединения, но эти идентификационные номера не обязательно должны совпадать. Идентификационный номер соединения имеет значение только для инструкций PROFINET в пользовательской программе соответствующего CPU.
- Для идентификатора соединения CPU можно использовать любой номер. Однако назначение ID соединений последовательно начиная с "1" обеспечивает простой метод для отслеживания числа используемых соединений для определенного CPU.

Примечание

Каждая инструкция TSEND_C, TRCV_C или TCON в программе пользователя создает новое соединение. Важно, чтобы для каждого соединения использовался бы правильный идентификатор соединения.

Следующий пример показывает коммуникации между двумя CPU, которые используют два отдельных соединения для отправки и получения данных.

- Инструкция TSEND_C в CPU_1 относится к инструкции TRCV_C в CPU_2 через первое соединение ("Идентификатор соединения 1" для обоих CPU, CPU_1 и CPU_2).
- Инструкция TRCV_C в CPU_1 относится к инструкции TSEND_C в CPU_2 через второе соединение ("Идентификатор соединения 2" для обоих CPU, CPU_1 и CPU_2).



① TSEND_C в CPU_1 создает соединение и присваивает этому соединению идентификатор (идентификатор соединения 1 для CPU_1).

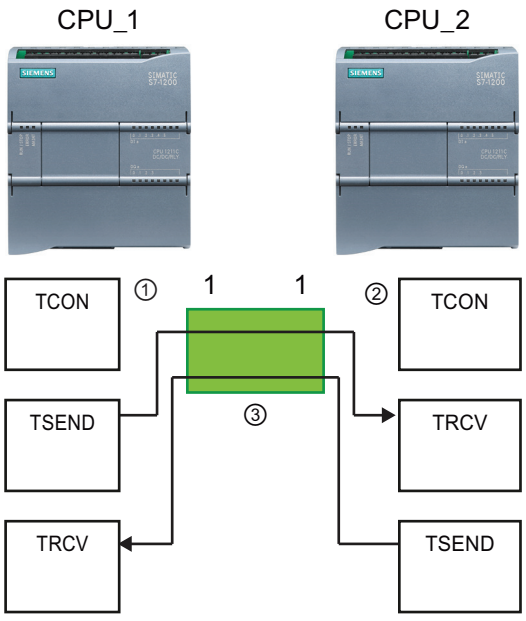
② TRCV_C в CPU_2 создает соединение для CPU_2 и присваивает идентификатор (идентификатор соединения 1 для CPU_2).

③ TRCV_C в CPU_1 создает второе соединение для CPU_1 и присваивает этому соединению другой идентификатор (идентификатор соединения 2 для CPU_1).

④ TSEND_C в CPU_2 создает второе соединение и присваивает этому соединению другой идентификатор (идентификатор соединения 2 для CPU_2).

Следующий пример показывает коммуникации между двумя CPU, которые используют только одно соединение для отправки и получения данных.

- Каждый CPU использует одну инструкция TCON для конфигурирования соединения между обоими CPU.
- Инструкция TSEND в CPU_1 относится к инструкции TRCV в CPU_2 через идентификатор соединения ("Идентификатор соединения 1"), который был сконфигурирован инструкцией TCON в CPU_1). Инструкция TRCV в CPU_2 относится к инструкции TSEND в CPU_1 через идентификатор соединения ("Идентификатор соединения 1"), который был сконфигурирован инструкцией TCON в CPU_2).
- Инструкция TSEND в CPU_2 относится к инструкции TRCV в CPU_1 через идентификатор соединения ("Идентификатор соединения 1"), который был сконфигурирован инструкцией TCON в CPU_2). Инструкция TRCV в CPU_1 относится к инструкции TSEND в CPU_2 через идентификатор соединения ("Идентификатор соединения 1"), который был сконфигурирован инструкцией TCON в CPU_1).



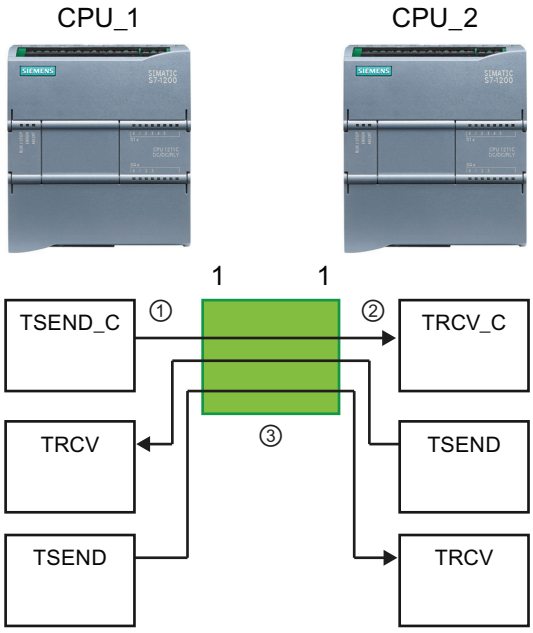
① Инструкция TCON в CPU_1 создает соединение и присваивает этому соединению идентификатор в CPU_1 (ID = 1).

② Инструкция TCON в CPU_2 создает соединение и присваивает этому соединению идентификатор в CPU_2 (ID = 1).

③ Инструкции TSEND и TRCV в CPU_1 используют созданный TCON в CPU_1 идентификатор соединения (ID = 1).

Инструкции TSEND и TRCV в CPU_2 используют созданный TCON в CPU_2 идентификатор соединения (ID = 1).

Как видно из следующего примера, и с помощью отдельных инструкций TSEND и TRCV возможна коммуникации через созданное инструкцией TSEND_C или TRCV_C соединение. Инструкции TSEND и TRCV сами не создают нового соединения, поэтому необходимо использовать DB и идентификатор соединения, созданные инструкцией TSEND_C, TRCV_C или TCON.



① Инструкция TSEND_C в CPU_1 создает соединение и присваивает этому соединению идентификатор (ID = 1).

② Инструкция TRCV_C в CPU_2 создает соединение и присваивает этому соединению идентификатор в CPU_2 (ID = 1).

③ Инструкции TSEND и TRCV в CPU_1 используют созданный TSEND_C в CPU_1 идентификатор соединения (ID = 1).

Инструкции TSEND и TRCV в CPU_2 используют созданный TRCV_C в CPU_2 идентификатор соединения (ID = 1).

11.2.8.6 Параметры для соединения PROFINET

Инструкциям TSEND_C, TRCV_C и TCON для подключения к партнерскому устройству необходимы специальные параметры соединения. Эти параметры назначаются структурой TCON_Param для протоколов TCP, ISO-on-TCP и UDP. Обычно эти параметры указываются в "Свойствах" инструкции на вкладке "Конфигурация" (Страница 812). Если доступ к вкладке "Конфигурация" отсутствует, то структура TCON_Param должна быть указана в параметрах инструкции.

В V4.1 или выше структура TCON_IP_v4 настраивает протокол TCP, а структура TCON_IP_RFC настраивает протокол ISO-on-TCP.

От V4.3 структура TCON_IP_V4_SEC назначает прочие параметры для протокола TCP. Чтобы установить безопасную связь TCP между двумя CPU S7-1200, необходимо создать блок данных с типом системных данных TCON_IP_V4_SEC или в каждом CPU, выполнить присвоение параметров и вызвать его непосредственно в инструкции. Инструкции TCON, TSEND_C и TRCV_C поддерживаю тип системных данных TCON_IP_V4_SEC.

От версии V4. 4 структуры TCON_QDN и TCON_QDN_SEC используются для настройки коммуникационных соединений для TCP и UDP через полное доменное имя, а также структура TCON_QDN_SEC для настройки коммуникационных соединений для TCP через полное доменное имя с защищенной связью.

TCON_Param

Таблица 11-8 Структура описания соединения (TCON_Param)

Байт	Параметр и тип данных		Описание
От 0 до 1	block_length	UInt	Длина: 64 байта (фикс.)
От 2 до 3	id	CONN_OUC (Word)	Ссылка на это соединение: Диапазон значений: от 1 (по умолчанию) до 4095. Указать значение этого параметра для инструкции TSEND_C, TRCV_C или TCON в пункте ID.
4	connection_type	USInt	Тип соединения: • 17: TCP (по умолчанию) • 18: ISO-on-TCP • 19: UDP
5	active_est	Bool	ID типа соединения: • TCP и ISO-on-TCP: – FALSE: пассивное соединение – TRUE: активное соединение (по умолчанию) • UDP: FALSE
6	local_device_id	USInt	ID локального интерфейса PROFINET или Industrial Ethernet: 1 (по умолчанию)
7	local_tsap_id_len	USInt	Длина параметра local_tsap_id в байтах; возможные значения: • TCP: 0 (активный, по умолчанию) или 2 (пассивный) • ISO-on-TCP: От 2 до 16 • UDP: 2
8	rem_subnet_id_len	USInt	Этот параметр не используется.

Байт	Параметр и тип данных		Описание
9	rem_staddr_len	USInt	Длина адреса конечной точки партнера в байтах: • 0: не указано (параметр rem_staddr не имеет значения) • 4 (по умолчанию): действительный IP-адрес в параметре rem_staddr (только для TCP и ISO-on-TCP)
10	rem_tsap_id_len	USInt	Длина параметра rem_tsap_id в байтах; возможные значения: • TCP: 0 (пассивный) или 2 (активный, по умолчанию) • ISO-on-TCP: От 2 до 16 • UDP: 0
11	next_staddr_len	USInt	Этот параметр не используется.
От 12 до 27	local_tsap_id	Array [1..16] of Byte	Компонент локального адреса соединения: • TCP и ISO-on-TCP: номер локального порта (возможные значения: от 1 до 49151; рекомендуемые значения: 2000...5000): – local_tsap_id[1] = старший байт номера порта в шестнадцатеричном представлении; – local_tsap_id[2] = младший байт номера порта в шестнадцатеричном представлении; – local_tsap_id[3-16] = неприменимо • ISO-on-TCP: локальный TSAP-ID: – local_tsap_id[1] = B#16#E0; – local_tsap_id[2] = монтажная стойка и слот локальных конечных точек (биты от 0 до 4: номер слота, биты 5 – 7: номер монтажной стойки); – local_tsap_id[3-16] = расширение TSAP, опция • UDP: Этот параметр не используется. Примечание: Убедиться в том, что каждое значение local_tsap_id является уникальным в CPU.
От 28 до 33	rem_subnet_id	Array [1..6] of USInt	Этот параметр не используется.
От 34 до 39	rem_staddr	Array [1..6] of USInt	Только для TCP и ISO-on-TCP: IP-адрес конечной точки партнера. (Неприменимо для пассивных соединений.) Пример: IP-адрес 192.168.002.003 сохраняется в следующие элементы массива: rem_staddr[1] = 192 rem_staddr[2] = 168 rem_staddr[3] = 002 rem_staddr[4] = 003 rem_staddr[5-6] = неприменимо

Байт	Параметр и тип данных		Описание
От 40 до 55	rem_tsap_id	Array [1..16] of Byte	Компонент партнерского адреса соединения: • TCP: Номер порта партнерского CPU. Диапазон: от 1 до 49151; рекомендуемые значения: от 2000 до 5000): – rem_tsap_id[1] = старший байт номера порта в шестнадцатеричном представлении; – rem_tsap_id[2] = младший байт номера порта в шестнадцатеричном представлении; – rem_tsap_id[3-16] = неприменимо • ISO-on-TCP: идентификатор TSAP партнера: – rem_tsap_id[1] = B#16#E0 – rem_tsap_id[2] = стойка и слот конечной точки партнера (биты 0 – 4: номер слота, биты 5 – 7: номер монтажной стойки) – rem_tsap_id[3-16] = TSAP расширение, опция • UDP: Этот параметр не используется.
От 56 до 61	next_staddr	Array [1..6] of Byte	Этот параметр не используется.
От 62 до 63	spare	Word	Зарезервировано: W#16#0000

TCON_IP_V4

Таблица 11-9 Структура описания соединения (TCON_IP_V4): для использования с TCP

Байт	Параметр и тип данных		Описание
От 0 до 1	InterfaceId	HW_ANY	Аппаратный идентификатор субмодуля IE-интерфейса
От 2 до 3	ID	CONN_OUC (Word)	Ссылка на это соединение: Диапазон значений: от 1 (по умолчанию) до 4095. Указать значение этого параметра для инструкции TSEND_C, TRCV_C или TCON в пункте ID.
4	ConnectionType	Byte	Тип соединения: • 11: TCP/IP (по умолчанию) • 17: TCP/IP (Этот тип соединения присутствует для подключения устаревших систем. Рекомендуется использование "11: TCP/IP (по умолчанию)".) • 19: UDP
5	ActiveEstablished	Bool	Активное/пассивное установление соединения: • TRUE: активное соединение (по умолчанию) • FALSE: пассивное соединение
	V4 IP-адрес		
6	ADDR[1]	Byte	Октет 1
7	ADDR[1]	Byte	Октет 2
8	ADDR[1]	Byte	Октет 3
9	ADDR[1]	Byte	Октет 4

Байт	Параметр и тип данных		Описание
От 10 до 11	RemotePort	UInt	Номер удаленного UDP/TCP порта
От 12 до 13	LocalPort	UInt	Номер локального UDP/TCP порта

TCON_IP_V4_SEC

Таблица 11-10 Структура описания соединения (TCON_IP_V4_SEC): для использования с TCP

Байт	Параметр и тип данных		Описание
От 0 до 15	ConnPara	TCON_IP_v4	SDT для параметров соединения Информация об идентификаторе интерфейса (interface_id): • Если оставить для interface_id предустановленное значение 0, то операционная система CPU анализирует удаленный IP-адрес и локальные IP-маршруты, а затем указывает интерфейс Industrial Ethernet модуля CPU для установления безопасного OUC соединения. В этом случае диагностические данные всегда назначаются первому интерфейсу Industrial Ethernet модуля CPU. • Если аппаратный идентификатор интерфейса Industrial Ethernet модуля CPU или CP указывается в качестве interface_id, то защищенное соединение OUC устанавливается через соответствующий интерфейс Industrial Ethernet.
16	ActivateSecure-Conn	Bool	Активация безопасной коммуникации для этого соединения Если этот параметр имеет значение FALSE (по умолчанию), последующие параметры обеспечения безопасности не имеют значения, т.е. соединение не является защищенным. В этом случае можно установить незащищенное соединение TCP или UDP.
17	TLSServerReq-ClientCert	Bool	Только для стороны сервера: Запрос сертификата X.509-V3 от TLS клиента. FALSE (по умолчанию)
От 18 до 19	ExtTLSCapabilities	Word	• Бит 0: Только для стороны клиента. Установленный бит означает, что клиент проверяет альтернативное имя субъекта сертификата (subjectAlternateName) в сертификате X.509-V3 сервера, чтобы проверить подлинность сервера. Сертификаты проверяются, когда соединение установлено. 16#0 (по умолчанию) • Биты от 1 до 15: Зарезервировано для будущих обновлений

Байт	Параметр и тип данных		Описание
От 20 до 23	TLSServerCertRef	UDInt	• Сторона сервера: ID собственного сертификата X.509-V3 • Сторона клиента: ID сертификата X.509-V3 (обычно сертификат CA), который используется клиентом TLS для проверки подлинности сервера TLS. Если этот параметр равен 0, клиент TLS использует все сертификаты (CA), загруженные в настоящее время в хранилище сертификатов клиента, для проверки подлинности сервера. 0 (по умолчанию)
От 24 до 27	TLSClientCertRef	UDInt	• Сторона клиента: ID собственного сертификата X.509-V3 • Сторона сервера: ID сертификата X.509-V3 (или группы сертификатов X.509-V3), который используется сервером TLS для проверки клиента TLS. Если этот параметр равен 0, сервер TLS использует для проверки подлинности клиента все сертификаты (CA), загруженные в настоящее время в хранилище сертификатов сервера. 0 (по умолчанию)

Параметр соединения CONNECT блоков данных экземпляров для инструкций TCON, TSEND_C и TRCV_C содержит ссылку на используемый блок данных.

Примечание

Можно установить незащищенные TCP- или UDP-соединения через IPv4.

Также можно использовать SDT TCON_IP_V4_SEC для незащищенного TCP- или UDP-соединения через IPv4.

TCON_IP_RFC

Таблица 11-11 Структура описания соединения (TCON_IP_RFC): Для использования с ISO-on-TCP

Байт	Параметр и тип данных		Описание
От 0 до 1	InterfaceId	HW_ANY	Аппаратный идентификатор субмодуля IE-интерфейса
От 2 до 3	ID	CONN_OUC (Word)	Ссылка на это соединение: Диапазон значений: от 1 (по умолчанию) до 4095. Указать значение этого параметра для инструкции TSEND_C, TRCV_C или TCON в пункте ID.
4	ConnectionType	Byte	Тип соединения: • 12: ISO-on-TCP (по умолчанию) • 17: ISO-on-TCP (Этот тип соединения присутствует для подключения устаревших систем. Рекомендуется использование "12: ISO-on-TCP (по умолчанию)".)
5	ActiveEstablished	Bool	Активное/пассивное установление соединения: • TRUE: активное соединение (по умолчанию) • FALSE: пассивное соединение
От 6 до 7	Резерв		Не используется
	V4 IP-адрес		
8	ADDR[1]	Byte	Октет 1
9	ADDR[1]	Byte	Октет 2

Байт	Параметр и тип данных		Описание
10	ADDR[1]	Byte	Октет 3
11	ADDR[1]	Byte	Октет 4
	Выбор удаленной передачи		
От 12 до 13	TSelLength	UInt	Длина TSelector
От 14 до 45	TSel	array [1..32] of Byte	Символьный массив для имени TSAP
	Выбор локальной передачи		
От 46 до 47	TSelLength	UInt	Длина TSelector
От 48 до 79	TSel	array [1..32] of Byte	Символьный массив для имени TSAP

TCON_QDN

Таблица 11-12 Структура описания соединения в соответствии с TCON_QDN

Байт	Параметр и тип данных		Описание
От 0 до 1	InterfaceId	HW_ANY	S7-1200 CPU от версии прошивки V4.4: • InterfaceId вставленного CP: – С CP для S7-1200 от версии прошивки V3.2 – С CP для ET 200SP от версии прошивки V2 S7-1200 CPU до версии V4.4: • Параметр не имеет значения
От 2 до 3	ID	CONN_OUC	Ссылка на это соединение (диапазон значений: от 1 до 4095). Указать значение этого параметра для инструкции TCON в ID.
4	ConnectionType	BYTE	Тип соединения: • 11: TCP (11 дес. = 0x0B шестн.) • 19: UDP (19 дес. = 0x13 шестн.)
5	ActiveEstablished	BOOL	Идентификатор для типа создания соединения: • FALSE: Пассивное создание соединения • TRUE: Активное создание соединения
От 6 до 261	RemoteQDN	Массив из STRING [1..254]	Полное доменное имя конечной точки партнера, которое должно заканчиваться на "." Следует помнить, что в сети SIMATIC имя, включая заключительную точку, не должно превышать 254 символа.
От 262 до 263	RemotePort	UINT	Адрес порта удаленного участника соединения
От 264 до 265	LocalPort	UINT	Адрес порта локального участника соединения

TCON_QDN_SEC

Таблица 11-13 Структура описания соединения согласно TCON_QDN_SEC

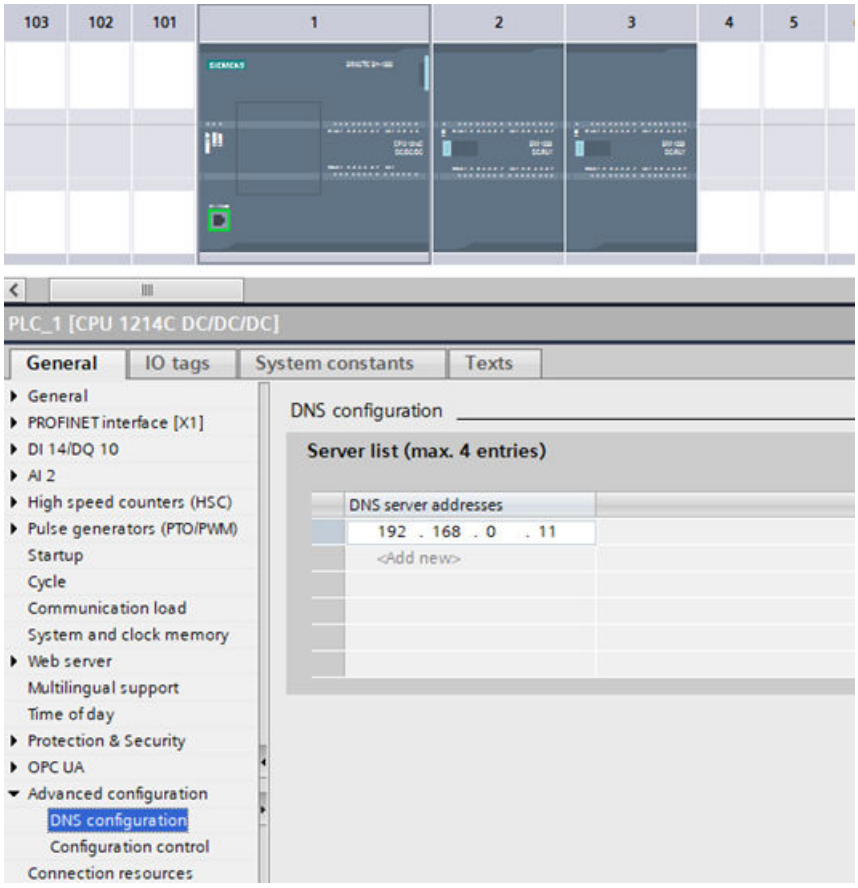
Байт	Параметр и тип данных		Описание
От 0 до 271	ConnPara	TCON_QDN	Параметры соединения
272	ActivateSecureConn	BOOL	Активация безопасной коммуникации для этого соединения. Если этот параметр имеет значение FALSE, последующие параметры обеспечения безопасности не имеют значения. В этом случае можно установить незащищенное соединение TCP или UDP.
273	TLSServerReqClientCert	BOOL	Только на стороне сервера: Запрос сертификата X.509-V3 от TLS клиента
От 274 до 275	ExtTLSCapabilities	WORD	• Бит 0: Только для стороны клиента. Установленный бит означает, что клиент проверяет альтернативное имя эмитента сертификата Subject Alternate Name) в сертификате X.509-V3 сервера, чтобы проверить подлинность сервера. Сертификаты проверяются, когда соединение установлено. • Биты от 1 до 15: зарезервировано для будущих обновлений
От 276 до 279	TLSServerCertRef	UDINT	• На стороне сервера: ID собственного сертификата X.509-V3 • На стороне клиента: ID сертификата X.509-V3 (обычно сертификат CA), который используется клиентом TLS для проверки подлинности сервера TLS. Если этот параметр равен 0, клиент TLS использует все сертификаты (CA), загруженные в настоящее время в хранилище сертификатов клиента, для проверки подлинности сервера.
От 280 до 283	TLSClientCertRef	UDINT	• На стороне клиента: ID собственного сертификата X.509-V3 • На стороне сервера: ID сертификата X.509-V3 (или группы сертификатов X.509-V3), который используется сервером TLS для проверки подлинности клиента TLS. Если этот параметр равен 0, сервер TLS использует для проверки подлинности клиента все сертификаты (CA), загруженные в настоящее время в хранилище сертификатов сервера.

11.2.8.7 Конфигурирование DNS

Для использования безопасных Открытых коммуникационных соединений пользователя (OUC) необходимо сконфигурировать систему доменных имен (DNS) . В сети должен находиться как минимум один DNS-сервер и следует сконфигурировать как минимум один DNS-сервер для S7-1200 CPU.

DNS-сервер конфигурируется следующим образом:

1. Перейти к просмотру устройств для S7-1200 CPU.
2. Открыть страницу "Свойства" и вкладку "Общее".
3. Кликнуть по конфигурации DNS для отображения страницы настройки.
4. Кликнуть в таблице со списком серверов в первой строке под адресами DNS-серверов по <Добавить новый> и ввести IP-адрес своего DNS-сервера.



11.2.8.8 Инструкции TSEND_C и TRCV_C

Версия V4.1 модуля S7-1200 CPU в комбинации с STEP 7 V13 SP1 или выше расширяет возможности инструкций TSEND_C иTRCV_C, позволяя использовать параметры соединения со структурами TCON_IP_V4 иTCON_IP_RFC.

Версия V4.3 модуля S7-1200 CPU в комбинации с STEP 7 V15.1 или выше расширяет возможности инструкций TSEND_C иTRCV_C, позволяя использовать параметры соединения со структурами TCON_IP_V4, TCON_IP_V4_SEC и TCON_IP_RFC.

Версия V4.4 модуля S7-1200 CPU в комбинации с STEP 7 V16 или выше расширяет возможности инструкций TSEND_C иTRCV_C, позволяя использовать параметры соединения со структурами TCON_IP_V4, TCON_IP_V4_SEC, TCON_IP_RFC, TCON_QDN и TCON_QDN_SEC.

По этой причине S7-1200 поддерживает два набора инструкций TSEND_C и TRCV_C:

- Старые инструкции TSEND_C и TRCV_C (Страница 856): Эти инструкции TSEND_C и TRCV_C использовались до версии V4.0 модуля S7-1200 и работают только с параметрами соединения со структурами по TCON_Param.
- Инструкции TSEND_C и TRCV_C (Страница 843): Эти инструкции TSEND_C и TRCV_C предлагают все функции старых инструкций и дополнительную возможность для использования параметров соединения со структурами по TCON_IP_V4, TCON_IP_V4_SEC, TCON_IP_RFC, TCON_QDN, and TCON_QDN_SEC.

Выбор версии инструкций TSEND_C и TRCV_C

Существует две версии инструкций TSEND_C и TRCV_C в STEP 7:

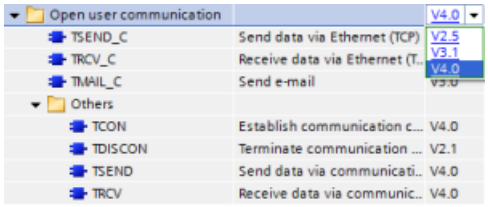
- Версии 2.5 и 3.1 были доступны в STEP 7 Basic/Professional V13 и более ранних версий.
- Версия 4.0 доступна в STEP 7 Basic/Professional V13 SP1 и более поздних версий.

Учитывая совместимость и для упрощения миграции можно выбирать, какая версия инструкций будет вставлена в программу пользователя.

Не следует никогда использовать инструкции разных версий в одной программе CPU.



Кликнуть по символу в окне задач с деревом инструкций, чтобы активировать заголовки и столбцы в дереве инструкций.



Для изменения версии инструкции TSEND_C и TRCV_C, выбрать соответствующую версию в раскрывающемся списке. Можно выбрать группу или отдельные инструкции.

Если инструкция TSEND_C или TRCV_C размещается через дерево инструкций в программе, то, в зависимости от выбранной инструкции TSEND_C или TRCV_C, в дереве проекта создается новый экземпляр FB или FC. Для просмотра нового экземпляра FB или FC открыть в дереве проекта PLC_x > Программные блоки > Системные блоки > Программные ресурсы.

Чтобы узнать версию инструкции TSEND_C или TRCV_C в программе, следует вызвать свойства в дерева проекта, а не свойства блочного элемента на экране в редакторе текстов программ. Выбрать в дереве проекта экземпляр FB или FC инструкции TSEND_C или TRCV_C, кликнуть правой кнопкой мыши, выбрать "Свойства" и после страницу "Информация", чтобы отобразить номер версии инструкции TSEND_C или TRCV_C.

TSEND_C и TRCV_C (передача и получение данных через Ethernet)

Инструкция TSEND_C объединяет функции инструкций TCON, TDISCON и TSEND . Инструкция TRCV_C объединяет функции инструкций TCON, TDISCON и TRCV. (Дополнительную информацию о данных инструкциях можно найти в разделе "TCON, TDISCON, TSEND и TRCV (Страница 864)".)

Минимальный размер данных, которые можно передать (TSEND_C) или получить (TRCV_C), составляет один байт; максимальный размер данных составляет 8192 байта.

TSEND_C не поддерживает передачу данных из булевых адресов, и TRCV_C не принимает данные в булевы адреса. Дополнительную информация о передаче данных с помощью этих инструкций можно найти в разделе с информацией о целостности данных (Страница 191).

Примечание

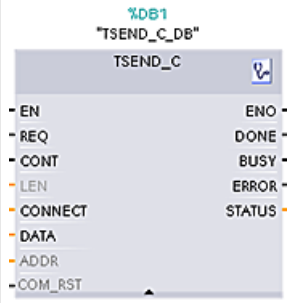
Инициализация параметров коммуникации

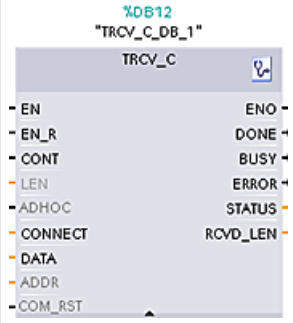
После вставки инструкции TSEND_C или TRCV_C, следует сконфигурировать в "Свойствах" инструкции (Страница 812) коммуникационные параметры (Страница 835). Если параметры для участников процесса коммуникации вводятся в окне инспектора, STEP 7 вводит соответствующие данные в DB инструкции.

Если будет использоваться мультиэкземплярный DB, то DB должен быть сконфигурирован вручную на обоих CPU.

Таблица 11-14 Инструкции TSEND_C и TRCV_C

LAD/FBD	SCL	Описание

	`"TSEND_C_DB"(`
	`req:=_bool_in_,`
	`cont:=_bool_in_,`
	`len:=_uint_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`connect:=_struct_inout_,`
	`data:=_variant_inout_,`
	`com_rst:=_bool_inout_);`
		TSEND_C устанавливает TCP или ISO on TCP соединение с партнером, отправляет данные и может завершить соединение. После настройки и установления соединения оно автоматически поддерживается и контролируется CPU.

	`"TRCV_C_DB"(`
	`en_r:=_bool_in_,`
	`cont:=_bool_in_,`
	`len:=_uint_in_,`
	`adhoc:=_bool_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`rcvd_len=>_uint_out_,`
	`connect:=_struct_inout_,`
	`data:=_variant_inout_,`
	`com_rst:=_bool_inout_);`
		TRCV_C устанавливает TCP или ISO on TCP соединение с CPU, принимает данные и может завершить соединение. После настройки и установления соединения оно автоматически поддерживается и контролируется CPU.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Таблица 11-15 Типы данных для параметров TSEND_C и TRCV_C

Параметр и тип данных		**Тип данных**	**Описание**
REQ (TSEND_C)	IN	Bool	Запускает операцию передачи по положительному фронту.
EN_R (TRCV_C)	IN	Bool	Активировать прием
CONT	IN	Bool	Управляет коммуникационным соединением: • 0: Отключение коммуникационного соединения после отправки данных. • 1: Установка и поддержание коммуникационного соединения При передаче данных (TSEND_C) (положительный фронт в параметре REQ) или при приеме данных (TRCV_C) (положительный фронт в параметре EN_R), параметр CONT должен иметь значение TRUE, чтобы установить или сохранить соединение.
LEN	IN	UDInt	Опциональный параметр (скрыт) Максимальное количество байтов, которые будут отправлены (TSEND_C) или получены (TRCV_C) при операции. Если в параметре DATA используются исключительно символьные значения, то параметр LEN должен иметь значение 0.
ADHOC (TRCV_C)	IN	Bool	Опциональный параметр (скрыт) Запрос Ad hoc режима для типа соединения TCP.

Параметр и тип данных		Тип данных	Описание
CONNECT	IN_OUT	Variant	Указатель на описание соединения: • Для TCP или UDP следует использовать структуру TCON_IP_v4 или TCON_QDN. Описание см.: "Параметры для соединения PROFINET (Страница 835)". • Для TCP с безопасной коммуникацией следует использовать структуру TCON_IP_V4_SEC или TCON_QDN_SEC. Описание см.: "Параметры для соединения PROFINET (Страница 835)". • Для ISO-on-TCP следует использовать структуру TCON_IP_RFC. Описание см.: "Параметры для соединения PROFINET (Страница 835)". • Для ISO соединений CP 1543-1 / CP 1545-1 следует использовать структуру TCON_ISOnative. Описание см. систему интерактивной помощи в TIA Portal: "Структура описания соединения согласно TCON_ISOnative". • Для соединений с клиентами SMS следует использовать тип системных данных TCON_PHONE. Описание см. систему интерактивной помощи в TIA Portal: "Параметры соединения по TCON_Phone". • Для FDL соединений CM 1542-5 следует использовать тип системных данных TCON_FDL. См. систему интерактивной помощи в TIA Portal: "Параметры соединения по TCON_FDL".
DATA	IN_OUT	Variant	Указатель на область передачи с: • Адрес и длина данных для передачи (TSEND_C) • Адрес и максимальная длина получаемых данных (TRCV_C)
ADDR	IN_OUT	Variant	Опциональный параметр (скрыт) Указатель на адрес получателя при типе соединения UDP. Информация об адресах назначается в структуре TADDR_Param ###.
COM_RST	IN_OUT	Bool	Опциональный параметр (скрыт) Перезапускает инструкцию: • 0: Не имеет значения • 1: Полностью перезапускает инструкцию. Имеющееся соединение либо завершается, либо сбрасывается и устанавливается заново согласно CONT. Параметр COM_RST сбрасывается после анализа с помощью инструкции TSEND_C или TRCV_C и поэтому не должен подключаться в статическом режиме.

Параметр и тип данных		Тип данных	Описание
DONE	OUT	Bool	Параметры состояния со следующими значениями: • 0: Операция передачи еще не была запущено или еще выполняется. • 1: Операция передачи была выполнена без ошибок. Это состояние отображается только в течение одного цикла.
BUSY	OUT	Bool	Параметры состояния со следующими значениями: • 0: Операция передачи еще не была запущена или уже завершена. • 1: Операция передачи еще не завершена. Новая операция передачи не может быть запушена.
ERROR	OUT	Bool	Параметры состояния со следующими значениями: • 0: Ошибки отсутствуют • 1: Ошибка при установлении соединения, передаче данных или завершении соединения.
STATUS	OUT	Word	Состояние инструкции (см. описание параметров ERROR и STATUS).
RCVD_LEN (TRCV_C)	OUT	Int	Фактический объем принятых данных (в Байтах).

Примечание

Для запуска операции передачи инструкции TSEND_C требуется переход 0 на 1 во входном параметре REQ. Параметр BUSY при обработке устанавливается на 1. Завершение операции передачи обозначается установкой одного из параметров DONE или ERROR на 1 на один цикл. В течение этого промежутка времени переход 0 на 1 во входном параметре REQ игнорируется.

Примечание

Настройка по умолчанию параметра LEN (LEN = 0) на основе параметра DATA определяет длину передаваемых данных. Рекомендуется, чтобы передаваемые инструкцией TSEND_C данные имели тот же размер, что и параметр DATA инструкции TRCV_C.

Если для параметра LEN используется настройка по умолчанию и данные должны передаваться партиями меньшего размера, чем параметр DATA, то действуют следующие правила. Если размер передаваемых TSEND_C данных не соответствует размеру параметра TRCV_C DATA, то TRCV_C остается в состоянии "Занято" (код состояния: 7006) до тех пор, пока общий размер передаваемых TSEND_C данных не сравняется с параметром TRCV_C DATA.

Буфер параметра TRCV_C DATA показывает новые полученные данные только после того, как размер данных сравняется с размером буфера параметра DATA.

Принцип работы TSEND_C

Инструкция TSEND_C выполняется асинхронно и последовательно реализует следующие функции:

1. Настройка и установка коммуникационного соединения:
   TSEND_C настраивает и устанавливает коммуникационное соединение, если в параметре REQ определяется положительный фронт и коммуникационное соединение еще отсутствует. После настройки и установления соединения оно автоматически поддерживается и контролируется CPU. Для настройки коммуникационного соединения используется заданное в параметре CONNECT описание соединения. Могут использоваться следующие типы соединений:
   - Структура TCON_Param для протоколов TCP, ISO-on-TCP и UDP
   - Для V4.1 и выше, TCP/UDP: Описание соединения с помощью структуры TCON_IP_V4 в параметре CONNECT
   - Для V4.1 и выше, ISO-on-TCP: Описание соединения с помощью структуры TCON_IP_RFC в параметре CONNECT
   - Для V4.3 и выше, TCP: Описание соединения с помощью структуры TCON_IP_V4_SEC в параметре CONNECT
   - Для V4.4 и выше, TCP: Описание соединения с помощью структур TCON_QDN и TCON_QDN_SEC в параметре CONNECT

   При переходе CPU в режим STOP текущее соединение завершается, а настроенное соединение удаляется. Для повторной настройки и установки соединения потребуется заново выполнить TSEND_C. Информацию о числе возможных коммуникационных соединений можно найти в технических характеристиках CPU.
2. Передача данных через существующее коммуникационное соединение:
   Операция передачи выполняется по положительному фронту в параметре REQ. Как описано выше, сначала создается коммуникационное соединение. Область передачи задается в параметре DATA. Она содержит адрес и длину передаваемых данных. Не использовать в параметре DATA область данных с типом данных BOOL или Array of BOOL. С помощью параметра LEN указывается максимальное число передаваемых за одну операцию байт. Если в параметре DATA используются символьное имя, то параметр LEN должен иметь значение 0.
   Данные, которые будут отправлены, не должны редактироваться, пока операция передачи не завершена.
3. Завершение коммуникационного соединения:
   Коммуникационное соединение разрывается после отправки данных, если параметр CONT к моменту положительного фронта в параметре REQ имел значение "0". Иначе, коммуникационное соединение будет поддерживаться.

При успешном выполнении операции передачи параметр DONE устанавливается на "1". Коммуникационное соединение может быть завершено заранее (см. вышеупомянутое описание зависимости от параметра CONT). Значение сигнала "1" в параметре DONE не служит подтверждением того, что отправленные данные уже были считаны участником коммуникации.

TSEND_C сбрасывается, если параметр COM_RST устанавливается на "1". Это может привести к потере данных, если данные передаются в данный момент.

В зависимости от параметра CONT возможны следующие сценарии:

- CONT = "0":
  Существующее коммуникационное соединение устанавливается.
- CONT = "1" и одно коммуникационное соединение было установлено:
  Существующее коммуникационное соединение сбрасывается и устанавливается заново.
- CONT = "1" и коммуникационное соединение не было установлено.
  Коммуникационное соединение не устанавливается.

Параметр COM_RST сбрасывается после обработки инструкцией T_SEND. Для повторной активации TSEND_C после выполнения (DONE = 1) следует однократно вызвать инструкцию с помощью REQ = 0.

Принцип работы TRCV_C

Инструкция TRCV_C выполняется асинхронно и последовательно реализует следующие функции:

1. Настройка и установка коммуникационного соединения:
   TRCV_C настраивает и устанавливает коммуникационное соединение, если параметр EN_R = "1" и коммуникационное соединение отсутствует. После настройки и установления соединения оно автоматически поддерживается и контролируется CPU. Для настройки коммуникационного соединения используется заданное в параметре CONNECT описание соединения. Могут использоваться следующие типы соединений:
   - Структура TCON_Param для протоколов TCP, ISO-on-TCP и UDP
   - Для V4.1 и выше, TCP/UDP: Описание соединения с помощью структуры TCON_IP_V4 в параметре CONNECT
   - Для V4.1 и выше, ISO-on-TCP: Описание соединения с помощью структуры TCON_IP_RFC в параметре CONNECT
   - Для V4.3 и выше, TCP: Описание соединения с помощью структуры TCON_IP_V4_SEC в параметре CONNECT
   - Для V4.4 и выше, TCP: Описание соединения с помощью структур TCON_QDN и TCON_QDN_SEC.

   При переходе CPU в режим STOP текущее соединение завершается, а настроенное соединение удаляется. Для повторной настройки и установки соединения потребуется заново выполнить TRCV_C с EN_R = 1.
   Если EN_R перед установкой коммуникационного соединения сбрасывается на 0, то соединение устанавливается и сохраняется, даже если CONT = 0. Однако прием данных не выполняется (DONE остается равным "0").
   Информацию о числе возможных коммуникационных соединений можно найти в технических характеристиках CPU.
2. Получение данных через существующее коммуникационное соединение:
   Если значение параметра EN_R равно "1", активируется получение данных. Как описано выше, сначала создается коммуникационное соединение. Полученные данные заносятся в область приема. Длина области приема задается в зависимости от используемого варианта протокола либо с помощью параметра LEN (еслиLEN <> 0), либо с помощью информации о длине параметра DATA (если LEN = 0). Если в параметре DATA используются исключительно символьные значения, то параметр LEN должен иметь значение 0.
   Если EN_R устанавливается на "0 до первого приема данных, то коммуникационное соединение сохраняется и при CONT = 0. Однако прием данных не выполняется (DONE остается равным "0").
3. Завершение коммуникационного соединения:
   Коммуникационное соединение разрывается после получения данных, если параметр CONT при установлении соединения имел значение "0". Иначе, коммуникационное соединение будет поддерживаться.

При успешном выполнении операции приема параметр DONE устанавливается на "1". Коммуникационное соединение может быть завершено заранее (см. вышеупомянутое описание зависимости от параметра CONT).

TRCV_C сбрасывается, если устанавливается параметр COM_RST. Если при повторном выполнении выполняется прием данных, то данные могут быть потеряны. В зависимости от параметра CONT возможны следующие сценарии:

- CONT = "0":
  Существующее коммуникационное соединение устанавливается.
- CONT = "1" и одно коммуникационное соединение было установлено:
  Существующее коммуникационное соединение сбрасывается и устанавливается заново.
- CONT = "1" и коммуникационное соединение не было установлено:
  Коммуникационное соединение не устанавливается.

Параметр COM_RST сбрасывается после обработки инструкцией TRCV_.

TRCV_C работает с теми же режимами приема, что и инструкция TRCV. В приведенной ниже таблице представлено, как данные записываются в область приема:

Вариант протокола	Доступность данных в области приема	Параметр Connection_type описания соединения	Параметр LEN	Параметр RCVD_LEN
TCP (Ad hoc режим)	Данные доступны сразу же.	B#16#11	Выбор с входом ADHOC инструкции TRCV_C	От 1 до 1472
TCP (получение данных с указанной длиной)	Данные доступны, если данные приняты полностью с указанной в параметре LEN длиной данных.	B#16#11	От 1 до 8192	Идентично значению в параметре LEN
ISO on TCP (передача данных с управлением по протоколу)	Данные доступны, если данные приняты полностью с указанной в параметре LEN длиной данных.	B#16#12	От 1 до 8192	Идентично значению в параметре LEN

Примечание

Ad hoc режим

"Ad hoc режим" доступен только для протокола TCP. Для конфигурирования инструкции TRCV_C для Ad hoc режима, установить входной параметр ADHOC инструкции. Длина области приема определяется при помощи указателя в параметре DATA. Действительно полученная длина данных выводится в параметре RCVD_LEN. Максимальный объем принимаемых данных составляет 1460 байт.

Примечание

Импорт STEP 7 проектов S7-300/400 с Ad hoc режимом в S7-1200

В STEP 7 проектах S7-300/400 Ad hoc режим устанавливается путем присвоения параметру LEN значения "0". Для конфигурирования инструкции TRCV_C для Ad hoc режима в S7-1200, установить входной параметр ADHOC инструкции.

При импорте STEP 7 проекта S7-300/400, включающего в себя Ad hoc режим, в S7-1200, необходимо изменить параметр LEN на 65535.

Примечание

TCP (получение данных с указанной длиной)

Длина для принимаемых данных задается значением параметра LEN. Данные, указанные в параметре DATA, доступны в области приема, если была полностью получена длина, указанная в параметре LEN.

Примечание

ISO on TCP (передача данных с управлением по протоколу)

При варианте протокола ISO on TCP данные передаются с управлением по протоколу. Область приема определяется параметрами LEN и DATA.

Параметры BUSY, DONE и ERROR

Примечание

Из-за асинхронной обработки TSEND_C данные в области передачи должны оставались непротиворечивыми, пока один из параметров DONE или ERROR не примет значение TRUE.

Для TSEND_C статус TRUE параметра DONE означает, что данные были успешно переданы. Это не означает, что партнерский CPU действительно считывает буфер приема.

Вследствие асинхронной обработки TRCV_C данные в области приема непротиворечивы только в том случае, если параметр DONE = 1.

Таблица 11-16 Параметры BUSY, DONE и ERROR инструкций TSEND_C и TRCV_C

BUSY	DONE	ERROR	Описание
1	0	0	Операция передачи обрабатывается.
0	1	0	Операция передачи успешно выполнена.
0	0	1	Установление соединения или операция передачи завершены с ошибкой. Причина ошибки зафиксирована в параметре STATUS.
0	0	0	Новая операция передачи не была назначена.

Состояние выполнения можно проверить с помощью параметров BUSY, DONE, ERROR и STATUS. Параметр BUSY показывает состояние обработки. С помощью параметра DONE можно проверить, успешно ли выполнена операция передачи. Параметр ERROR устанавливается, если во время выполнения инструкции TSEND_C или TRCV_C возникли ошибки. Информация об ошибках выводится в параметре STATUS.

Параметры Error иStatus

Таблица 11-17 Коды условий TSEND_C иTRCV_C для ERROR и STATUS

ERROR	STATUS * (W#16#...)	Описание
0	0000	Операция передачи (TSEND_C) или операция приема (TRCV_C) выполнены без ошибок.
0	0001	Коммуникационное соединение установлено.
0	0003	Коммуникационное соединение закрыто.
0	7000	Нет активной операции передачи; коммуникационное соединение не установлено.
0	7001	• Запустить выполнение операции передачи (TSEND_C) или операции приема (TRCV_C). • Установить соединение. • Ожидать участника соединения.
0	7002	Выполнение операции (REQ не имеет значения).
0	7003	Инструкция завершает коммуникационное соединение.
0	7004	Коммуникационное соединение установлено и контролируется; нет активной операции передачи (TSEND_C) или операции приема (TRCV_C).
0	7005	TSEND_C: Выполняется передача данных.
0	7006	TRCV_C: Инструкция принимает данные.
1	8085	• Параметр LEN больше, чем максимально допустимое значение. • Значение параметра LEN или DATA было изменено инструкцией после первого вызова.
1	8086	Параметр ID в параметре CONNECT находится вне допустимого диапазона.
1	8087	Достигнуто максимальное число соединений; дальнейшие соединения невозможны.
1	8088	Значение параметра LEN не соответствует области приема, указанной в параметре DATA.
1	8089	• Параметр CONNECT ссылается не на блок данных. • Параметр CONNECT ссылается не на описание соединения. • Созданное вручную описание соединения имеет неправильную структуру для выбранного типа соединения.
1	8091	Превышена максимальная глубина вложенности.
1	809A	Параметр CONNECT указывает на поле, не соответствующее длине описания соединения.
1	809B	InterfaceId в описании соединения не соответствует CPU или CP.
1	80A1	• Соединение или порт используются. • Ошибка обмена данными: – Указанное соединение еще не установлено. – Указанное соединение завершается. Передача через это соединение невозможна. – Интерфейс инициализируется заново.
1	80A2	Локальный или удаленный порт используется системой. Дополнительную информацию можно найти здесь: "Инструкции TCON и TDISCON" (Страница 864), "Коды условий ERROR и STATUS".".
1	80A3	• Попытка заново установить имеющееся соединение. • Попытка завершить несуществующее соединение. • Вложенная инструкция T_DIAG сообщает, что инструкция закрыла соединение.
1	80A4	Недопустимый IP-адрес удаленной конечной точки соединения, т. е. он совпадает с IP-адресом локального участника.

ERROR	STATUS * (W#16#...)	Описание
1	80A7	Ошибка обмена данными: Инструкция с COM_RST = 1 была вызвана прежде, чем была завершена операция передачи.
1	80AA	Другой блока устанавливает соединение с таким же ID соединения. Повторить операцию с новым положительным фронтом в параметре REQ.
1	80B3	• При использовании варианта протокола UDP параметры ADDR не содержит данных. • Ошибка в описании соединения • Другое описание соединения уже использует локальный порт.
1	80B4	При пассивном установлении соединения (ActiveEstablished = FALSE) с использованием варианта протокола ISO-on-TCP (ConnectionType = B#16#12) было нарушено одно или несколько из следующих условий: • local_tsap_id_len >= B#16#02 • local_tsap_id[1] = B#16#E0
1	80B5	Для типа соединения 13 = UDP допускается только пассивное установление соединения.
1	80B6	Ошибка настройки в параметре ConnectionType блока данных для описания соединения.
1	80B7	• Для типа системных данных TCON_Param: Ошибка в одном из следующих параметров блока данных для описания соединения: block_length, local_tsap_id_len, rem_subnet_id_len, rem_staddr_len, rem_tsap_id_len, next_staddr_len. • Для типов системных данных TCON_IP_V4, TCON_IP_RFC, TCON_IP_V4_SEC, TCON_QDN, и TCON_QDN_SEC : Инструкция установила IP-адрес конечной точки устройства-партнера на 0.0.0.0.
1	80C3	• Все ресурсы соединения заняты. • Блок с таким ID уже обрабатывается в другой группе приоритетов.
1	80C4	Временная ошибка обмена данными: • Инструкция в настоящий момент не может установить соединение. • Инструкция не может установить соединение, так как брандмауэры по пути соединения не открыты для требуемых портов. • Интерфейс находится в процессе приема новых параметров, либо инструкция устанавливает соединение. • Инструкция "TDISCON (Страница 864)" удаляет сконфигурированное соединение. • Вызов с COM_RST = 1 завершает используемое соединение. • У участника соединения временно нет ресурсов для приема. Участник соединения не готов к приему.
1	80C5	• Соединение разорвано коммуникационным партнером. • Удаленный участник соединения не разблокировал LSAP.
1	80C6	Ошибка сети: • Локальное устройство не может связаться с удаленным устройством-партнером. • Физическое прерывание на PROFIBUS
1	8722	Ошибка в параметре CONNECT: Недействительная исходная область (область не объявлена в блоке данных).
1	873A	Ошибка в параметре CONNECT: Доступ к описанию соединения невозможен (нет доступа к блоку данных).
1	877F	Ошибка в параметре CONNECT: Внутренняя ошибка

ERROR	STATUS * (W#16#...)	Описание
1	8822	TSEND_C:параметр DATA: Недопустимая исходная область, область отсутствует в DB.
1	8824	TSEND_C:параметр DATA: Ошибка области в указателе VARIANT.
1	8832	TSEND_C:параметр DATA: Слишком большой номер DB.
1	883A	TSEND_C:параметр CONNECT: Доступ к указанным данным соединения невозможен (например, так как отсутствует DB).
1	887F	TSEND_C:параметр DATA: Внутренняя ошибка (например, недопустимая ссылка на VARIANT)
1	893A	TSEND_C:параметр DATA: Доступ к области передачи невозможен (например, из-за отсутствия DB).
1	8922	TRCV_C:параметр DATA: Недопустимая целевая область, область отсутствует в DB.
1	8924	TRCV_C:параметр DATA: Ошибка области в указателе VARIANT.
1	8932	TRCV_C:параметр DATA: Слишком большой номер DB.
1	893A	TRCV_C:параметр CONNECT: Доступ к указанным данным соединения невозможен (например, так как отсутствует DB).
1	897F	TRCV_C:параметр DATA: Внутренняя ошибка (например, недопустимая ссылка на VARIANT).
1	8A3A	TRCV_C:параметр DATA: Отсутствует доступ к области данных (например, из-за отсутствия блока данных).
* Коды ошибок могут отображаться в редакторе текстов программ в виде целых или шестнадцатеричных значений.		

Примечание

Сообщения об ошибках инструкций TCON, TSEND, TRCV и TDISCON

Системное; инструкция TSEND_C использует инструкции TCON, TSEND и TDISCON, а инструкция TRCV_C использует инструкции TCON, TRCV и TDISCON. Дополнительную информацию по сообщениям об ошибках для данных инструкций можно найти в разделе TCON, TDISCON, TSEND и TRCV (Страница 864).

Протоколы соединения Ethernet

У каждого CPU есть встроенный порт PROFINET с поддержкой стандартной PROFINET коммуникации. Все инструкции TSEND_C и TRCV_C, а также TSEND и TRCV, поддерживают TCP и ISO on TCP протоколы Ethernet.

Дополнительную информацию см. в "Конфигурация устройства: Конфигурирование пути соединения между локальным и партнерским CPU (Страница 812)".

См. также

Идентификаторы (ID) для открытых коммуникационных соединений пользователя (Страница 832)

11.2.8.9 Инструкции TSEND_C и TRCV_C в старых системах

До STEP 7 V13 SP1 и S7-1200 V4.1-CPU, инструкции TSEND_C и TRCV_C работали только с параметрам соединения со структурами по "TCON_Param". Общие концепции применимы к обоим типам инструкций. См. описания отдельные старых инструкций TSEND_C и TRCV_C для получения информации о программировании.

Выбор версии инструкций TSEND_C и TRCV_C

Существует две версии инструкций TSEND_C и TRCV_C в STEP 7:

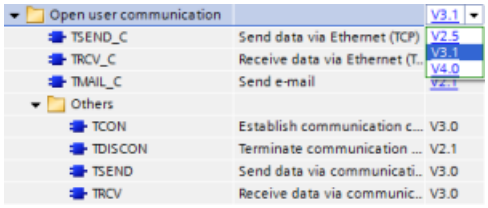
- Версии 2.5 и 3.1 были доступны в STEP 7 Basic/Professional V13 и более ранних версий.
- Версия 4.0 доступна в STEP 7 Basic/Professional V13 SP1 и более поздних версий.

Учитывая совместимость и для упрощения миграции можно выбирать, какая версия инструкций будет вставлена в программу пользователя.

Не следует никогда использовать инструкции разных версий в одной программе CPU.



Кликнуть по символу в окне задач с деревом инструкций, чтобы активировать заголовки и столбцы в дереве инструкций.



Для изменения версии инструкции TSEND_C и TRCV_C, выбрать соответствующую версию в раскрывающемся списке. Можно выбрать группу или отдельные инструкции.

Если инструкция TSEND_C или TRCV_C размещается через дерево инструкций в программе, то, в зависимости от выбранной инструкции TSEND_C или TRCV_C, в дереве проекта создается новый экземпляр FB или FC. Для просмотра нового экземпляра FB или FC открыть в дереве проекта PLC_x > Программные блоки > Системные блоки > Программные ресурсы.

Чтобы узнать версию инструкции TSEND_C или TRCV_C в программе, следует вызвать свойства в дерева проекта, а не свойства блочного элемента на экране в редакторе текстов программ. Выбрать в дереве проекта экземпляр FB или FC инструкции TSEND_C или TRCV_C, кликнуть правой кнопкой мыши, выбрать "Свойства" и после страницу "Информация", чтобы отобразить номер версии инструкции TSEND_C или TRCV_C.

Старые инструкции TSEND_C и TRCV_C (передача и прием данных через Ethernet)

Старая инструкция TSEND_C объединяет функции старых инструкций TCON, TDISCON и TSEND . Инструкция TRCV_C объединяет функции инструкций TCON, TDISCON и TRCV. (Дополнительную информацию о данных инструкциях можно найти в разделе "Старые инструкции TCON, TDISCON, TSEND и TRCV (TCP коммуникация) (Страница 876)".)

Минимальный размер данных, которые можно передать (TSEND_C) или получить (TRCV_C), составляет один байт; максимальный размер данных составляет 8192 байта. TSEND_C не поддерживает передачу данных из булевых адресов, и TRCV_C не принимает данные в булевы адреса. Дополнительную информация о передаче данных с помощью

этих инструкций можно найти в разделе с информацией о целостности данных (Страница 191).

Примечание

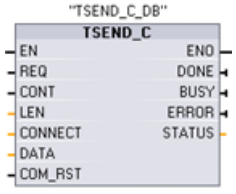
Инициализация параметров коммуникации

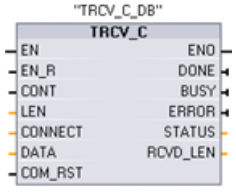
После вставки инструкции TSEND_C или TRCV_C, следует сконфигурировать в "Свойствах" инструкции (Страница 812) коммуникационные параметры (Страница 835). Если параметры для участников процесса коммуникации вводятся в окне инспектора, STEP 7 вводит соответствующие данные в DB инструкции.

Если будет использоваться мультиэкземплярный DB, то DB должен быть сконфигурирован вручную на обоих CPU.

Таблица 11-18 Инструкции TSEND_C и TRCV_C

LAD/FBD	SCL	Описание

	`"TSEND_C_DB"(`
	`req:=_bool_in_,`
	`cont:=_bool_in_,`
	`len:=_uint_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`connect:=_struct_inout_,`
	`data:=_variant_inout_,`
	`com_rst:=_bool_inout_);`
		TSEND_C устанавливает TCP или ISO on TCP соединение с партнером, отправляет данные и может завершить соединение. После настройки и установления соединения оно автоматически поддерживается и контролируется CPU.

	`"TRCV_C_DB"(`
	`en_r:=_bool_in_,`
	`cont:=_bool_in_,`
	`len:=_uint_in_,`
	`adhoc:=_bool_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`rcvd_len=>_uint_out_,`
	`connect:=_struct_inout_,`
	`data:=_variant_inout_,`
	`com_rst:=_bool_inout_);`
		TRCV_C устанавливает TCP или ISO on TCP соединение с CPU, принимает данные и может завершить соединение. После настройки и установления соединения оно автоматически поддерживается и контролируется CPU.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Таблица 11-19 Типы данных для параметров TSEND_C и TRCV_C

Параметр и тип данных		Тип данных	Описание
REQ (TSEND_C)	IN	Bool	REQ = 1 запускает операцию передачи TSEND_C по положительному фронту через описанное в параметре CONNECT соединение. CONT = 1 также необходима для установления и поддержания коммуникационного соединения.
EN_R (TRCV_C)	IN	Bool	Если EN_R = 1, тоTRCV_C готова к приему. Операция приема обрабатывается. CONT = 1 также необходима для установления и поддержания коммуникационного соединения.
CONT	IN	Bool	Управляет коммуникационным соединением: • 0: Разрыв коммуникационного соединения • 1: Установка и поддержание коммуникационного соединения При передаче данных (TSEND_C) (положительный фронт в параметре REQ), параметр CONT должен иметь значение TRUE, чтобы можно было установить или сохранить соединение. При приеме данных (TRCV_C) (положительный фронт в параметре EN_R), параметр CONT должен иметь значение TRUE, чтобы можно было установить или сохранить соединение.
LEN	IN	UInt	Максимальное количество байтов, которые должны быть отправлены (TSEND_C) или получены (TRCV_C): • По умолчанию = 0: Параметр DATA определяет длину передаваемых (TSEND_C) или принимаемых (TRCV_C) данных. • Ad hoc режим = 65535: Для приема данных установлена переменная длина (TRCV_C).
CONNECT	IN_OUT	TCON_Param	Указатель на описание соединения (Страница 835)
DATA	IN_OUT	Variant	• Содержит адрес и длину данных для передачи (TSEND_C). • Содержит адрес и максимальную длину принимаемых данных (TRCV_C).
COM_RST	IN_OUT	Bool	Обеспечивает перезапуск инструкции: • 0: Не имеет значения • 1: Полный перезапуск функционального блока, существующее соединение завершается.
DONE	OUT	Bool	• 0: Операция еще не запущена или еще выполняется. • 1: Операция выполнена без ошибок.
BUSY	OUT	Bool	• 0: Операция завешена. • 1: Операция еще не завершена. Новая операция не может быть запушена.
ERROR	OUT	Bool	Параметры состояния со следующими значениями: • 0: Ошибки отсутствуют • 1: При обработке возникла ошибка. STATUS содержит информацию о типе ошибки.

Параметр и тип данных		Тип данных	Описание
STATUS	OUT	Word	Информация о состоянии включает в себя информацию об ошибке. (См. таблицу "Параметры ошибок и состояния" ниже.)
RCVD_LEN (TRCV_C)	OUT	Int	Фактический объем принятых данных в байтах

Примечание

Для запуска операции передачи инструкции TSEND_C требуется переход 0 на 1 во входном параметре REQ. Параметр BUSY при обработке устанавливается на 1. Завершение операции передачи обозначается установкой одного из параметров DONE или ERROR на 1 на один цикл. В течение этого промежутка времени переход 0 на 1 во входном параметре REQ игнорируется.

Примечание

Настройка по умолчанию параметра LEN (LEN = 0) на основе параметра DATA определяет длину передаваемых данных. Рекомендуется, чтобы передаваемые инструкцией TSEND_C данные имели тот же размер, что и параметр DATA инструкции TRCV_C.

Если для параметра LEN используется настройка по умолчанию и данные должны передаваться партиями меньшего размера, чем параметр DATA, то действуют следующие правила. Если размер передаваемых TSEND_C данных не соответствует размеру параметра DATA из TRCV_C, то TRCV_C остается в состоянии "Занято" (код состояния: 7006) до тех пор, пока общий размер передаваемых TSEND_C данных не сравняется с параметромDATA из TRCV_C.

Буфер параметра DATA из TRCV_C показывает новые полученные данные только после того, как размер данных сравняется с размером буфера параметра DATA.

Принцип работы TSEND_C

Следующие функции описывают принцип действия инструкции TSEND_C:

- Для установления соединения необходимо выполнить TSEND_C с CONT = 1.
- После успешной установки соединения, TSEND_C устанавливает параметр DONE на один цикл.
- Для завершения соединения, выполнить TSEND_C с CONT = 0. После соединение сразу же отменяется. Это относится и к принимающему устройству. Соединение завершается, и данные в буфере приема могут быть потеряны.
- Для отправки данных по существующему соединению, TSEND_C должна быть выполнена при положительном фронте на REQ. После успешной передачи, TSEND_C устанавливает параметр DONE на 1 на один цикл.
- Для установки соединения и передачи данных, TSEND_C должна быть выполнена с CONT = 1 и REQ = 1. После успешной передачи, TSEND_C устанавливает параметр DONE на 1 на один цикл.

Принцип работы TRCV_C

Следующие функции описывают принцип действия инструкции TRCV_C:

- Для установления соединения необходимо выполнить TRCV_C с CONT = 1.
- Для приема данных, выполнить TRCV_C с параметром EN_R = 1. TRCV_C непрерывно принимает данные, если параметры EN_R = 1 и CONT = 1.
- Для завершения соединения, выполнить TRCV_C с CONT = 0. Соединение сразу же завершается, и данные могут быть потеряны.

TRCV_C работает с теми же режимами приема, что и инструкция TRCV. В приведенной ниже таблице представлено, как данные записываются в область приема:

Таблица 11-20 Запись данных в область приема

Вариант протокола	Ввод данных в области приема	Параметр "connection_type"	Значение параметра LEN	Значение параметра RCVD_LEN (в байтах)
TCP	Ad hoc режим	B#16#11	65535	От 1 до 1472
TCP	Получение данных с указанной длиной	B#16#11	0 (рекомендуется) или от 1 до 8192, кроме 65535	От 1 до 8192
ISO on TCP	Ad hoc режим	B#16#12	65535	От 1 до 1472
ISO on TCP	Управляется протоколом	B#16#12	0 (рекомендуется) или от 1 до 8192, кроме 65535	От 1 до 8192

Примечание

Ad hoc режим

Ad hoc режим доступен только для вариантов протокола TCP и ISO on TCP. Ad hoc режим можно установить, присвоив параметру LEN значение "65535". Область приема идентична указанной параметром DATA области. Длина полученных данных выводится в параметре RCVD_LEN.

Если данные хранятся в "оптимизированном" DB (только символьная адресация), то данные могут быть приняты только в массивах данных типа Byte, Char, USInt иSInt.

Примечание

Импорт STEP 7 проектов S7-300/400 с Ad hoc режимом в S7-1200

В STEP 7 проектах S7-300/400 Ad hoc режим устанавливается путем присвоения параметру LEN значения "0". Для S7-1200 можно установить Ad hoc режим, присвоив параметру LEN значение "65535".

При импорте STEP 7 проекта S7-300/400, включающего в себя Ad hoc режим, в S7-1200, необходимо изменить параметр LEN на 65535.

Примечание

Данные в области передачи должны оставались непротиворечивыми, пока один из параметров DONE или ERROR не примет значение TRUE.

Из-за асинхронной обработки TSEND_C данные в области передачи должны оставались непротиворечивыми, пока один из параметров DONE или ERROR не примет значение TRUE.

Для TSEND_C статус TRUE параметра DONE означает, что данные были успешно переданы. Это не означает, что партнерский CPU действительно считывает буфер приема.

Вследствие асинхронной обработки TRCV_C данные в области приема непротиворечивы только в том случае, если параметр DONE = 1.

Таблица 11-21 Параметры BUSY, DONE и ERROR инструкций TSEND_C и TRCV_C

BUSY	DONE	ERROR	Описание
TRUE	Не имеет значения	Не имеет значения	Операция находится в обработке.
FALSE	TRUE	FALSE	Операция успешно выполнена.
FALSE	FALSE	TRUE	Операция завершена с ошибкой. Причина ошибки зафиксирована в параметре STATUS.
FALSE	FALSE	FALSE	Не была назначена новая операция.

Параметры Error иStatus

Таблица 11-22 Коды условий TSEND_C иTRCV_C для ERROR и STATUS

ERROR	STATUS	Описание
0	0000	Операция выполнена без ошибок
0	7000	Нет активных операций
0	7001	Запустить обработку операции, установить соединение, ожидать участника соединения
0	7002	Данные передаются или принимаются
0	7003	Соединение завершается
0	7004	Соединение установлено, выполняется мониторинг, нет активных операций
1	8085	Параметр LEN больше, чем максимально допустимое значение.
1	8086	Параметр CONNECT находится вне допустимого диапазона.
1	8087	Достигнуто максимальное число соединений; дальнейшие соединения невозможны.
1	8088	Параметр LEN не подходит для указанной в DATA области памяти.
1	8089	Параметр CONNECT ссылается не на блок данных.
1	8091	Превышена максимальная глубина вложенности.
1	809A	Параметр CONNECT указывает на поле, не соответствующее длине описания соединения.

ERROR	STATUS	Описание
1	809B	local_device_id в описании соединения не соответствует CPU.
1	80A1	Ошибка обмена данными: • Указанное соединение еще не установлено. • Указанное соединение в настоящий момент завершается; передача данных через это соединение невозможна • Интерфейс инициализируется заново
1	80A3	Попытка завершить несуществующее соединение
1	80A4	Недействительный IP-адрес удаленного устройства. Пример: IP-адрес удаленного устройства совпадает с IP-адресом локального устройства.
1	80A5	ID соединения (Страница 832) уже используется.
1	80A7	Ошибка обмена данными: Вызов TDISCON до того, как было завершено выполнение TSEND_C.
1	80B2	Параметр CONNECT указывает на блок данных, который был сгенерирован с ключевым словом UNLINKED.
1	80B3	Противоречивые параметры: • Ошибка в описании соединения • Локальный интерфейс (параметр local_tsap_id) уже встречается в другом описании соединения. • ID в описании соединения не совпадает с указанным в качестве параметра ID.
1	80B4	При установке пассивного соединения через ISO on TCP (connection_type = B#16#12), код состояния 80B4 предупреждает о том, что введенный TSAP не соответствовал одному из следующих требований к адресу: • Для длины локального TSAP равной 2 и значения идентификатора E0 или E1 (в шестнадцатеричном формате) для первого байта, второй байт должен быть 00 или 01. • Для длины локального TSAP равной 3 или больше и значения идентификатора E0 или E1 (в шестнадцатеричном формате) для первого байта, второй байт должен быть 00 или 01, а все остальные байты должны быть допустимыми ASCII-символами. • Для длины локального TSAP равной 3 или больше и первого байта значения идентификатора TSAP, не имеющего значения E0 или E1 (в шестнадцатеричном формате), все байты TSAP ID должны быть допустимыми ASCII-символами. Действительные ASCII-символы представляют собой байтовые значения от 20 до 7E (в шестнадцатеричном формате).
1	80B7	Тип данных и/или длина переданных данных не подходят для области в партнерском CPU, в которую необходимо выполнить запись данных.
1	80C3	Все ресурсы соединения заняты.
1	80C4	Временная ошибка обмена данными: • В настоящий момент соединение не может быть установлено • Интерфейс находится в процессе приема новых параметров • Сконфигурированное соединение удаляется TDISCON.
1	8722	Параметр CONNECT: Недействительная исходная область: Область отсутствует в DB.
1	873A	Параметр CONNECT: Доступ к описанию соединения невозможен (например, DB отсутствует)

ERROR	STATUS	Описание
1	877F	Параметр CONNECT: Внутренняя ошибка, например, недопустимая ссылка на ANY.
1	893A	Параметр содержит номер блока данных, который не загружен.

Протоколы соединения Ethernet

У каждого CPU есть встроенный порт PROFINET с поддержкой стандартной PROFINET коммуникации. Все инструкции TSEND_C и TRCV_C, а также TSEND и TRCV, поддерживают TCP и ISO on TCP протоколы Ethernet.

Дополнительную информацию см. в "Конфигурация устройства: Конфигурирование пути соединения между локальным и партнерским CPU (Страница 812)".

11.2.8.10 Инструкции TCON, TDISCON, TSEND и TRCV

От версии V4.1 модуля S7-1200 CPU в комбинации с STEP 7 V13 SP1 или выше расширяются возможности инструкции TCON, позволяя использовать параметры соединения со структурами по TCON_IP_V4 и TCON_IP_RFC. Для S7-1200 CPU возможности инструкций TSEND и TRCV теперь также позволяют использовать параметры соединения со структурами TCON_IP_V4 и TCON_IP_RFC.

От версии V4.3 модуля S7-1200 CPU в комбинации с STEP 7 V15.1 или выше расширяются возможности инструкции TCON, позволяя использовать параметры соединения со структурами по TCON_IP_V4, TCON_IP_V4_SEC и TCON_IP_RFC.

От версии V4.4 модуля S7-1200 CPU в комбинации с STEP 7 V13 V16 или выше расширяются возможности инструкции TCON, позволяя использовать параметры соединения со структурами по TCON_IP_V4, TCON_IP_V4_SEC, TCON_IP_RFC, TCON_QDN и TCON_QDN_SEC.

По этой причине S7-1200 поддерживает два набора инструкций TCON, TDISCON, TSEND и TRCV:

- Старые инструкции TCON, TDISCON, TSEND и TRCV (Страница 876): Эти инструкции TCON, TDISCON, TSEND и TRCV использовались до версии V4.0 модуля S7-1200 и работают только с параметрами соединения со структурами по TCON_Param.
- Инструкции TCON, TDISCON, TSEND и TRCV (Страница 864): Эти инструкции TCON, TDISCON, TSEND и TRCV предлагают все функции старых инструкций и дополнительную возможность для использования параметров соединения со структурами по TCON_IP_V4, TCON_IP_V4_SEC, TCON_IP_RFC, TCON_QDN и TCON_QDN_SEC.

Выбор версии инструкций TCON, TDISCON, TSEND и TRCV

Существует две версии инструкций TCON, TDISCON, TSEND или TRCV в STEP 7:

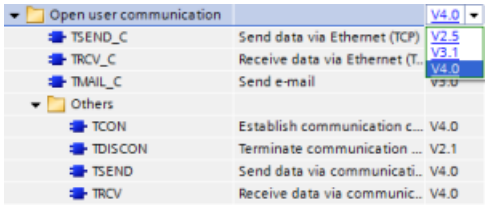
- Версии 2.5 и 3.1 были доступны в STEP 7 Basic/Professional V13 и более ранних версий.
- Версия 4.0 доступна в STEP 7 Basic/Professional V13 SP1 и более поздних версий.

Учитывая совместимость и для упрощения миграции можно выбирать, какая версия инструкций будет вставлена в программу пользователя.

Не следует никогда использовать инструкции разных версий в одной программе CPU.



Кликнуть по символу в окне задач с деревом инструкций, чтобы активировать заголовки и столбцы в дереве инструкций.



Для изменения версии инструкции TCON, TDISCON, TSEND или TRCV, выбрать соответствующую версию в раскрывающемся списке. Можно выбрать группу или отдельные инструкции.

Если инструкция TCON, TDISCON, TSEND или TRCV размещается через дерево инструкций в программе, то, в зависимости от выбранной инструкции TCON, TDISCON, TSEND или TRCV, в дереве проекта создается новый экземпляр FB или FC. Для просмотра нового экземпляра FB или FC открыть в дереве проекта PLC_x > Программные блоки > Системные блоки > Программные ресурсы.

Чтобы узнать версию инструкции TCON, TDISCON, TSEND или TRCV в программе, следует вызвать свойства в дерева проекта, а не свойства блочного элемента на экране в редакторе текстов программ. Выбрать в дереве проекта экземпляр FB или FC инструкции TCON, TDISCON, TSEND или TRCV, кликнуть правой кнопкой мыши, выбрать "Свойства" и после страницу "Информация", чтобы отобразить номер версии инструкции TCON, TDISCON, TSEND или TRCV.

Инструкции TCON, TDISCON, TSEND и TRCV (TCP коммуникация)

Ethernet коммуникация с протоколами TCP и ISO on TCP

Примечание

Инструкции TSEND_C и TRCV_C

Для упрощения программирования PROFINET/Ethernet коммуникации, инструкции TSEND_C и TRCV_C объединяют функциональность инструкций TCON, TDISCON. TSEND и TRCV:

- TSEND_C объединяет инструкции TCON, TDISCON и TSEND.
- TSEND_C объединяет инструкции TCON, TDISCON и TRCV.

Следующие инструкции управляют процессом коммуникации:

- Инструкция TCON устанавливает соединение TCP/IP между клиентским и серверным компьютером (CPU).
- Инструкции TSEND и TRCV передают и принимают данные.
- Инструкция TDISCON разрывает соединение.

Минимальный размер данных, которые можно передать (TSEND) или получить (TRCV), составляет один байт; максимальный размер данных составляет 8192 байта. TSEND не поддерживает передачу данных из булевых адресов, и TRCV не принимает данные в булевы адреса. Дополнительную информация о передаче данных с помощью этих

инструкций можно найти в разделе с информацией о целостности данных (Страница 191).

Инструкции TCON, TDISCON, TSEND и TRCV выполняются асинхронно, поэтому обработка одной операции происходит на несколько выполнений инструкции. Так, например, операция по настройке и установке соединения запускается путем выполнения инструкции TCON с параметром REQ = 1. После происходит мониторинг выполнения операции, для этого инструкция TCON выполняется повторно и проверяется, подтверждается ли параметр DONE завершение операции.

В следующей таблице представлены отношения между BUSY, DONE и ERROR: Текущее состояние операции можно определить на основании таблицы:

Таблица 11-23 Взаимодействие между параметрами BUSY, DONE и ERROR

BUSY	DONE	ERROR	Описание
1	0	0	Операция находится в обработке.
0	1	0	Операция успешно выполнена.
0	0	1	Операция завершена с ошибкой. Причина ошибки выводится в параметре STATUS.
0	0	0	Не была назначена новая операция.

TCON и TDISCON

Примечание

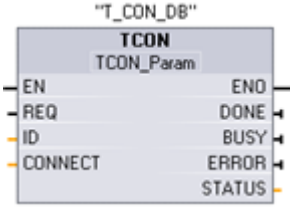
Инициализация параметров коммуникации

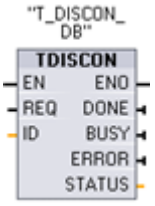
После вставки инструкции TCON, следует сконфигурировать в "Свойствах" инструкции (Страница 812) коммуникационные параметры (Страница 835). Если параметры для участников процесса коммуникации вводятся в окне инспектора, STEP 7 вводит соответствующие данные в блок данных экземпляра инструкции.

Если будет использоваться мультиэкземплярный DB, то DB должен быть сконфигурирован вручную на обоих CPU.

Таблица 11-24 Инструкции TCON и TDISCON

LAD/FBD		Описание

	`"TCON_DB"(`
	`req:=_bool_in_,`
	`ID:=_undef_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`connect:=_struct_inout_);`
		TCP и ISO on TCP: TCON инициирует коммуникационное соединение CPU с устройством-участником процесса коммуникации.

	`"TDISCON_DB"(`
	`req:=_bool_in_,`
	`ID:=_word_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_);`
		TCP и ISO on TCP: TDISCON завершает коммуникационное соединение CPU с устройством-участником процесса коммуникации.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Таблица 11-25 Типы данных для параметров TCON и TDISCON

Параметр	Объявление	Тип данных	Описание
REQ	IN	Bool	Запускает операцию по установлению соединения, указанного в ID, при положительном фронте.
ID	IN	CONN_OUC (Word)	Ссылка на назначенное соединение. Диапазон значений: W#16#0001 до W#16#0FFF

Параметр	Объявление	Тип данных	Описание
CONNECT (TCON)	IN_OUT	VARIANT	Указатель на описание соединения: • Для TCP или UDP следует использовать структуру TCON_IP_v4 или TCON_QDN. Описание см.: "Параметры для соединения PROFINET (Страница 835)". • Для TCP с безопасной коммуникацией следует использовать структуру TCON_IP_V4_SEC или TCON_QDN_SEC. Описание см.: "Параметры для PROFINET- (Страница 835)соединения (Страница 835)". • Для ISO-on-TCP следует использовать структуру TCON_IP_RFC. Описание см.: "Параметры для соединения PROFINET (Страница 835)". • Для ISO соединений CP 1543-1 / CP 1545-1 следует использовать структуру TCON_ISOnative. Описание см.: Система интерактивной помощи в TIA Portal: "Структура описания соединения согласно TCON_ISOnative". • Для соединений с клиентами SMS следует использовать тип системных данных TCON_PHONE. Описание см. систему интерактивной помощи в TIA Portal: "Параметры соединения по TCON_Phone". • Для FDL соединений CM 1542-5 следует использовать тип системных данных TCON_FDL. См. систему интерактивной помощи в TIA Portal: "Параметры соединения по TCON_FDL".
DONE	OUT	Bool	Параметры состояния со следующими значениями: • 0: Операция еще не запущена или еще выполняется. • 1: Операция выполнена без ошибок.
BUSY	OUT	Bool	Параметры состояния со следующими значениями: • 0: Операция передачи еще не была запущена или уже завершена. • 1: Операция еще не завершена. Новая операция не может быть запушена.
ERROR	OUT	Bool	Параметр состояния ERROR: • 0: Ошибки отсутствуют • 1: Возникла ошибка
STATUS	OUT	Word	Состояние инструкции

Оба коммуникационных партнера выполняют инструкцию TCON, чтобы настроить и установить коммуникационное соединение. С помощью параметров задаются активная и пассивная конечные точки коммуникации. После настройки и установки соединения оно автоматически поддерживается и контролируется CPU.

Если соединение завершается, например, из-за разрыва линии или удаленным участником, то активный участник пытается восстановить его. Нет необходимости повторно выполнять инструкцию TCON.

При выполнении инструкции TDISCON или при переходе CPU в режим STOP текущее соединение завершается, а настроенное соединение удаляется. Для повторной настройки и установки соединения потребуется заново выполнить TCON.

Таблица 11-26 Коды условий ERROR иSTATUS для TCON и TDISCON

ERROR	STATUS * (W#16#...)	Объяснение
0	0000	Соединение было успешно установлено.
0	7000	Нет активных операций
0	7001	Запустить обработку операции; установить соединение (TCON) или завершить соединение (TDISCON).
0	7002	Инструкция устанавливает соединение (REQ не имеет значения); установить соединение (TCON) или завершить соединение (TDISCON).
1	8085	TCON: ID соединения используется.
1	8086	TCON: Параметр ID находится вне допустимого диапазона.
1	8087	TCON: Достигнуто максимальное число соединений; дальнейшие соединения невозможны
1	8089	TCON: Параметр CONNECT указывает не на описание соединения, или описание соединения создано вручную.
1	809A	TCON: Структура в параметре CONNECT не поддерживается инструкцией или длина недействительная.
1	809B	TCON: • Элемент InterfaceId в описании соединения не соответствует CPU или CP, или равен "0". • Элемент InterfaceId в структуре TCON_xxx ссылается не на идентификатор оборудования CPU или CM/CP интерфейса.
1	80A1	TCON: Для TCP/UDP: Соединение или порт используются.
1	80A2	TCON: Система использует локальный или удаленный порт. Дополнительную информацию см. в "Общие параметры для инструкций" (Страница 930), "ОграниченныеTSAPs и номера портов для пассивной ISO и TCP коммуникации".
1	80A3	TCON: Созданное пользовательской программой соединение (TCON) использует значение в параметре ID. Соединение использует такой же ID и такие же настройки соединения в параметре CONNECT.
1	80A4	TCON: Недопустимый IP-адрес удаленной конечной точки соединения, или он совпадает с IP-адресом локального участника.
1	80A7	TCON: Ошибка обмена данными: Инструкция "TDISCON" выполнена до завершения "TCON".
1	80B3	Противоречивые параметры
1	80B4	TCON: Только с TCON_IP_RFC: Возникала одна из следующих ситуаций: • Инструкция не назначила локальный транспортный селектор. • Первый байт не содержит значение 0x0E. • Локальный транспортный селектор начинается с "SIMATIC-".
1	80B5	TCON: Инструкция допускает только пассивное установление соединения для типа соединения 13 = UDP (параметрActiveEstablished структуры TCON_xxx имеет значение TRUE).

ERROR	STATUS * (W#16#...)	Объяснение
1	80B6	TCON: Ошибка настройки в параметре ConnectionType блока данных для описания соединения: • Допустимо только для TCON_IP_V4, TCON_IP_V4_SEC, TCON_QDN, TCON_QDN_SEC: 0x11, 0x0B и 0x13 • Допустимо только для TCON_IP_RFC: 0x0C и 0x12
1	80B7	TCON: Для TCON_IP_V4, TCON_IP_V4_SEC, TCON_QDN, TCON_QDN_SEC: • TCP (активное установление соединения) Entfernter порт "0". • TCP (пассивное установление соединения): Локальный порт "0". • UDP: Локальный порт "0". • Инструкция установила IP-адрес конечной точки устройства-партнера на 0.0.0.0. TCON: Для TCON_IP_RFC: • Локальный (local_tselector) или удаленный (remote_tselector) транспортный селектор был указан инструкцией с длиной более 32 байт. • Для TSelLength транспортного селектора (локального или удаленного) инструкция назначила длина свыше 32. • Ошибка длины в IP-адресе определенного участника соединения • Инструкция установила IP-адрес конечной точки устройства-партнера на 0.0.0.0.
1	80B8	TCON: ID параметра в локальном описании соединения (структура в параметре CONNECT) и ID параметра в инструкции являются различными.
1	80C3	TCON: Все ресурсы соединения заняты.
1	80C4	Временная ошибка обмена данными: • Инструкция в настоящий момент не может установить соединение (TCON). • Инструкция не может установить соединение, так как брандмауэры по пути соединения не открыты для требуемых портов (TCON). • Интерфейс находится в процессе приема новых параметров (TCON и TDISCON). • Инструкция "TDISCON" удаляет сконфигурированное соединение (TCON).
1	80C5	TCON: Удаленный участник выполнил одну из следующих операций: • Он препятствовал установке соединения • Удалил соединение • Активно завершил соединение
1	80C6	TCON: Инструкция не может связаться с удаленным устройством-партнером (ошибка сети).
1	80C7	TCON: Превышение времени при выполнении
1	80C8	TCON: Созданное пользовательской программой соединение (TCON) использует значение в параметре ID. Соединение использует такой же ID, но другие настройки соединения в параметре CONNECT.
1	80C9	TCON: Подтверждение удаленного участника не удалось. Удаленный участник, который хочет установить соединение, не соответствует участнику, определенному в структуре параметра CONNECT.
1	80CE	TCON: IP-адрес локального интерфейса 0.0.0.0.
1	80E0	TCON: Инструкция выполнила прием непригодного или плохого сообщения.

ERROR	STATUS * (W#16#...)	Объяснение
1	80E1	TCON: Ошибка при подтверждении. Возможные причины: • Отмена со стороны пользователя • Недостаточный уровень безопасности. • Инструкция не поддерживает повторное согласование. • Инструкция не поддерживает версию SSL/TLS. • Валидация имени хост-системы не удалось.
1	80E2	Сертификат не поддерживается / недействительный сертификат / нет сертификата Возможная причина: CPU не установил время для затронутого модуля или модуль не синхронизирован. Пример: Стандартная установка для даты модуля 1/1/2012, CPU не установил дату при вводе в эксплуатацию. Срок действия сертификата начинается 20 августа 2016 г. и завершается 20 августа 2024 г. В этом случае дата модуля не соответствует сроку действия сертификата и поэтому сертификат считается недействительным для модуля.
1	80E3	Сертификат отклонен.
1	80E4	Не найдет действительный орган сертификации.
1	80E5	Сертификат истек.
1	80E6	Ошибки целостности в протоколе безопасности транспортного уровня
1	80E7	Не поддерживаемое расширение в сертификате X.509-V3
1	80E9	Инструкция не поддерживает сервер TLS без сертификата сервера.
1	80EA	Инструкция не поддерживает протокол DTLS (UDP).
1	80EB	Клиент не может запросить сертификат клиента.
1	80EC	Сервер не может выполнить валидацию на базе subjectAlternateName (это доступно только для клиентов).
1	80ED	TLSServerCertRef_m-ID недопустимо
*Коды ошибок могут отображаться в редакторе текстов программ в виде целочисленных или шестнадцатеричных значений.		

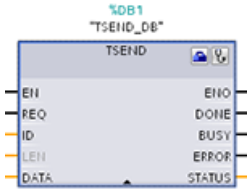
TSEND и TRCV

Примечание

При выполнении в открытых коммуникационных соединениях пользователя через PROFINET инструкции TSEND без соответствующей инструкции TRCV на удаленном устройстве, может случиться, что инструкция TSEND может оставаться неопределенно долго в состоянии "Busy", ожидая приема данных инструкцией TRCV. В этом состоянии выход "Busy" инструкции TSEND установлен и выход "Состояние" имеет значение "0x7002". Такое условие может возникнуть, если передается больше 4096 байт данных. Проблема устраняется при следующем выполнении инструкции TRCV.

Таблица 11-27 Инструкции TSEND и TRCV

LAD/FBD	SCL	Описание

	`"TSEND_DB"(`
	`req:=_bool_in_,`
	`ID:=_word_in_,`
	`len:=_udint_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`data:=_variant_inout_);`
		TCP и ISO on TCP: TSEND передает через коммуникационное соединение данные с CPU на станцию-партнер.

	`"TRCV_DB"(`
	`en_r:=_bool_in_,`
	`ID:=_word_in_,`
	`len:=_udint_in_,`
	`adhoc:=_bool_in_,`
	`ndr=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`rcvd_len=>_udint_out_,`
	`data:=_variant_inout_);`
		TCP и ISO on TCP: TRCV принимает через коммуникационное соединение данные от станции-партнера на CPU.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Таблица 11-28 Типы данных для параметров TSEND и TRCV

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	TSEND: Запускает операцию передачи по положительному фронту. Данные передаются из указанной с помощью DATA и LEN области.
EN_R	IN	Bool	TRCV: Активирует CPU для приема, если EN_R = 1, тоTRCV готова к приему. Операция приема обрабатывается.
ID	IN	CONN_OUC (Word)	Это ссылка на соответствующее соединение. ID должен совпадать с соответствующим ID параметра в локальном описании соединения. Диапазон значений: от W#16#0001 до W#16#0FFF
LEN	IN	UDInt	Максимальное количество байтов, которые должны быть отправлены (TSEND) или получены (TRCV): • По умолчанию = 0: Параметр DATA определяет длину передаваемых (TSEND) или принимаемых (TRCV) данных. • Ad hoc режим = 65535: Для приема данных установлена переменная длина (TRCV).
ADHOC	IN	Bool	TRCV: Опциональный параметр (скрыт) Запрос Ad hoc режима для типа соединения TCP.
DATA	IN_OUT	Variant	Указатель на область данных для передачи (TSEND) или приема (TRCV). Область данных содержит адрес и длину. Адрес относится к одной из областей памяти I, Q или M или к DB.

Параметр и тип данных		Тип данных	Описание
DONE	OUT	Bool	TSEND: • 0: Операция еще не запущена или еще выполняется. • 1: Операция выполнена без ошибок.
NDR	OUT	Bool	TRCV: • NDR = 0: Операция еще не запущена или еще выполняется. • NDR = 1: Операция была успешно выполнена.
BUSY	OUT	Bool	• BUSY = 1: Операция еще не завершена. Новая операция не может быть запушена. • BUSY = 0: Операция завешена.
ERROR	OUT	Bool	ERROR = 1: При обработке возникла ошибка. STATUS содержит информацию о типе ошибки.
STATUS	OUT	Word	Информация о состоянии включает в себя информацию об ошибке. (См. таблицу "Коды условий для ERROR и STATUS" ниже.)
RCVD_LEN	OUT	UDInt	TRCV: Фактический объем принятых данных в байтах

Примечание

Для запуска операции передачи инструкции TSEND требуется переход 0 на 1 во входном параметре REQ. Параметр BUSY при обработке устанавливается на 1. Завершение операции передачи обозначается установкой одного из параметров DONE или ERROR на 1 на один цикл. В течение этого промежутка времени переход 0 на 1 во входном параметре REQ игнорируется.

Принцип работы TRCV

Инструкция TRCV записывает принятые данные в область приема, которая определяется следующими двумя переменными:

- Определяет начало области
- Длина области или значение, предоставляемое на входе LEN, если он не 0

Примечание

Настройка по умолчанию параметра LEN (LEN = 0) на основе параметра DATA определяет длину передаваемых данных. Рекомендуется, чтобы передаваемые инструкцией TSEND данные имели тот же размер, что и параметр DATA инструкции TRCV.

Если для параметра LEN используется настройка по умолчанию и данные должны передаваться партиями меньшего размера, чем параметр DATA, то действуют следующие правила. Рекомендуется оставить бит EN_R на 1, пока соответствующая инструкция TSEND передаст соответствующий объем данных, чтобы заполнить параметр DATA из TRCV. Если размер передаваемых TSEND данных не соответствует размеру параметра DATA из TRCV, то TRCV остается в состоянии "Занято" (код состояния: 7002) до тех пор, пока бит EN_R = 1, пока общий размер передаваемых TSEND данных не сравняется с параметром DATA из TRCV. Если бит EN_R из TRCV принимает импульсы, то он должен получить столько импульсов, сколько раз должна быть выполнена инструкция TSEND для приема данных.

Буфер параметра DATA из TRCV показывает новые полученные данные только после того, как размер данных сравняется с размером буфера параметра DATA.

Как только все данные операции получены, они передаются из TRCV в область приема, и NDR устанавливается на 1.

Таблица 11-29 Запись данных в область приема

Вариант протокола	Ввод данных в области приема	Параметр "connection_type"	Значение параметра LEN	Значение параметра RCVD_LEN (в байтах)
TCP	Ad hoc режим	B#16#11	Выбор с входом ADHOC инструкции TRCV	От 1 до 1472
TCP	Получение данных с указанной длиной	B#16#11	0 (рекомендуется) или от 1 до 8192, кроме 65535	От 1 до 8192
ISO on TCP	Ad hoc режим	B#16#12	65535	От 1 до 1472
ISO on TCP	управляется протоколом	B#16#12	0 (рекомендуется) или от 1 до 8192, кроме 65535	От 1 до 8192

Примечание

Ad hoc режим

Ad hoc режим доступен только для вариантов протокола TCP и ISO on TCP. Для конфигурирования инструкции TRCV для Ad hoc режима, установить входной параметр ADHOC инструкции. Область приема идентична указанной параметром DATA области. Длина полученных данных выводится в параметре RCVD_LEN. Непосредственно после получения блока данных инструкция TRCV записывает данные в область приема и устанавливает NDR на "1".

Если данные хранятся в "оптимизированном" DB (только символьная адресация), то данные могут быть приняты только в массивах данных типа Byte, Char, USInt иSInt.

Примечание

Импорт STEP 7 проектов S7-300/400 с Ad hoc режимом в S7-1200

В STEP 7 проектах S7-300/400 Ad hoc режим устанавливается путем присвоения параметру LEN значения "0". Для конфигурирования инструкции TRCV для Ad hoc режима в S7-1200, установить входной параметр ADHOC инструкции.

При импорте STEP 7 проекта S7-300/400, включающего в себя Ad hoc режим, в S7-1200, необходимо изменить параметр LEN на 65535.

Таблица 11-30 Коды условий ERROR иSTATUS для TSEND и TRCV

ERROR	STATUS	Описание
0	0000	• Операция передачи была выполнена без ошибок (TSEND) • Новые данные приняты: Фактическая длина полученных данных отображается в RCVD_LEN (TRCV).
0	7000	• Нет активных операций (TSEND) • Блок не готов к получению (TRCV)
0	7001	• Запустить обработку операции, идет отправка данных: Во время обработки операционная система обращается к данным в области передачиDATA (TSEND). • Блок готов к приему, операция приема была активирована (TRCV).
0	7002	• Следующий вызов (REQ не имеет значения), операция находится в обработке: Во время обработки операционная система обращается к данным в области передачиDATA (TSEND). • Следующий вызов, операция приема обрабатывается: Во время обработки данные записываются в область приема. Поэтому возникновение ошибки может привести появлению противоречивых данных в области приема (TRCV).
1	8085	• Параметр LEN больше, чем максимально допустимое значение (TSEND) и (TRCV). • Параметр LEN или DATA изменился после последнего выполнения инструкции (TRCV).
1	8086	Параметр ID за пределами допустимой области адресов.
1	8088	Параметр LEN больше, чем указанная в DATA область памяти.

ERROR	STATUS	Описание
1	80A1	Ошибка обмена данными: • Указанное соединение еще не установлено (TSEND и TRCV). • Указанное соединение завершается. Операция передачи или приема по этому соединению невозможна (TSEND и TRCV). • Интерфейс инициализируется заново (TSEND). • Интерфейс находится в процессе приема новых параметров (TRCV).
1	80C3	Внутренний дефицит ресурсов соединения (Страница 832): Блок с таким ID уже обрабатывается в другом классе приоритета.
1	80C4	Временная ошибка обмена данными: • В настоящий момент соединение с партнером не может быть установлено • Интерфейс находится в процессе приема новых настроек для параметров, либо в данный момент выполняется установка соединения.

Протоколы соединения Ethernet

У каждого CPU есть встроенный порт PROFINET с поддержкой стандартной PROFINET коммуникации. Инструкции TSEND_C, TRCV_C, TSEND и TRCV поддерживают TCP и ISO on TCP протоколы Ethernet.

Дополнительную информацию см. в "Конфигурация устройства: Конфигурирование пути соединения между локальным и партнерским CPU (Страница 812)".

11.2.8.11 Старые инструкции TCON, TDISCON, TSEND и TRCV

До STEP 7 V13 SP1 и S7-1200 V4.1-CPU, инструкции TCON, TDISCON, TSEND и TRCV работали только с параметрам соединения со структурами по "TCON_Param". Общие концепции применимы к обоим типам инструкций. См. описания отдельные старых инструкций TCON, TDISCON, TSEND и TRCV для получения информации о программировании.

Выбор версии инструкций TCON, TDISCON, TSEND и TRCV

Существует две версии инструкций TCON, TDISCON, TSEND или TRCV в STEP 7:

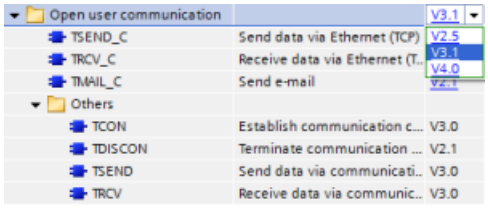
- Версии 2.5 и 3.1 были доступны в STEP 7 Basic/Professional V13 и более ранних версий.
- Версия 4.0 доступна в STEP 7 Basic/Professional V13 SP1 и более поздних версий.

Учитывая совместимость и для упрощения миграции можно выбирать, какая версия инструкций будет вставлена в программу пользователя.

Не следует никогда использовать инструкции разных версий в одной программе CPU.



Кликнуть по символу в окне задач с деревом инструкций, чтобы активировать заголовки и столбцы в дереве инструкций.



Для изменения версии инструкции TCON, TDISCON, TSEND или TRCV, выбрать соответствующую версию в раскрывающемся списке. Можно выбрать группу или отдельные инструкции.

Если инструкция TCON, TDISCON, TSEND или TRCV размещается через дерево инструкций в программе, то, в зависимости от выбранной инструкции TCON, TDISCON, TSEND или TRCV, в дереве проекта создается новый экземпляр FB или FC. Для просмотра нового экземпляра FB или FC открыть в дереве проекта PLC_x > Программные блоки > Системные блоки > Программные ресурсы.

Чтобы узнать версию инструкции TCON, TDISCON, TSEND или TRCV в программе, следует вызвать свойства в дерева проекта, а не свойства блочного элемента на экране в редакторе текстов программ. Выбрать в дереве проекта экземпляр FB или FC инструкции TCON, TDISCON, TSEND или TRCV, кликнуть правой кнопкой мыши, выбрать "Свойства" и после страницу "Информация", чтобы отобразить номер версии инструкции TCON, TDISCON, TSEND или TRCV.

Старые инструкции TCON, TDISCON, TSEND и TRCV (TCP коммуникация)

Ethernet коммуникация с протоколами TCP и ISO on TCP

Примечание

Инструкции TSEND_C и TRCV_C

Для упрощения программирования PROFINET/Ethernet коммуникации, инструкции TSEND_C и TRCV_C объединяют функциональность инструкций TCON, TDISCON. TSEND и TRCV:

- TSEND_C объединяет инструкции TCON, TDISCON и TSEND.
- TSEND_C объединяет инструкции TCON, TDISCON и TRCV.

Следующие инструкции управляют процессом коммуникации:

- Инструкция TCON устанавливает соединение TCP/IP между клиентским и серверным компьютером (CPU).
- Инструкции TSEND и TRCV передают и принимают данные.
- Инструкция TDISCON разрывает соединение.

Минимальный размер данных, которые можно передать (TSEND) или получить (TRCV), составляет один байт; максимальный размер данных составляет 8192 байта. TSEND не поддерживает передачу данных из булевых адресов, и TRCV не принимает данные в булевы адреса. Дополнительную информация о передаче данных с помощью этих

инструкций можно найти в разделе с информацией о целостности данных (Страница 191).

Инструкции TCON, TDISCON, TSEND и TRCV выполняются асинхронно, поэтому обработка одной операции происходит на несколько выполнений инструкции. Так, например, операция по настройке и установке соединения запускается путем выполнения инструкции TCON с параметром REQ = 1. После происходит мониторинг выполнения операции, для этого инструкция TCON выполняется повторно и проверяется, подтверждается ли параметр DONE завершение операции.

В следующей таблице представлены отношения между BUSY, DONE и ERROR: Текущее состояние операции можно определить на основании таблицы:

Таблица 11-31 Взаимодействие между параметрами BUSY, DONE и ERROR

BUSY	DONE	ERROR	Описание
TRUE	Не имеет значения	Не имеет значения	Операция находится в обработке.
FALSE	TRUE	FALSE	Операция успешно выполнена.
FALSE	FALSE	TRUE	Операция завершена с ошибкой. Причина ошибки зафиксирована в параметре STATUS.
FALSE	FALSE	FALSE	Не была назначена новая операция.

TCON и TDISCON

Примечание

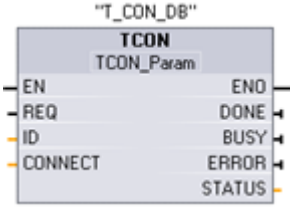
Инициализация параметров коммуникации

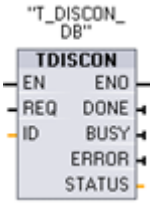
После вставки инструкции TCON, следует сконфигурировать в "Свойствах" инструкции (Страница 812) коммуникационные параметры (Страница 835). Если параметры для участников процесса коммуникации вводятся в окне инспектора, STEP 7 вводит соответствующие данные в блок данных экземпляра инструкции.

Если будет использоваться мультиэкземплярный DB, то DB должен быть сконфигурирован вручную на обоих CPU.

Таблица 11-32 Инструкции TCON и TDISCON

LAD/FBD		Описание

	`"TCON_DB"(`
	`req:=_bool_in_,`
	`ID:=_undef_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`connect:=_struct_inout_);`
		TCP и ISO on TCP: TCON инициирует коммуникационное соединение CPU с устройством-участником процесса коммуникации.

	`"TDISCON_DB"(`
	`req:=_bool_in_,`
	`ID:=_word_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_);`
		TCP и ISO on TCP: TDISCON завершает коммуникационное соединение CPU с устройством-участником процесса коммуникации.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Таблица 11-33 Типы данных для параметров TCON и TDISCON

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Управляющий параметр REQ запускает операцию путем установления указанного в ID соединения. Операция начинается по положительному фронту.
ID	IN	CONN_OUC (Word)	Ссылка на соединение с удаленным участником или между программой пользователя и сеансовым уровнем операционной системы, которое должно быть установлено (TCON) или разорвано (TDISCON). ID должен совпадать с соответствующим ID параметра в локальном описании соединения. Диапазон значений: от W#16#0001 до W#16#0FFF
CONNECT (TCON)	IN_OUT	TCON_Param	Указатель на описание соединения (Страница 835)
DONE	OUT	Bool	• 0: Операция еще не запущена или еще выполняется. • 1: Операция выполнена без ошибок.
BUSY	OUT	Bool	• 0: Операция завешена. • 1: Операция еще не завершена. Новая операция не может быть запушена.
ERROR	OUT	Bool	Параметры состояния со следующими значениями: • 0: Ошибки отсутствуют • 1: При обработке возникла ошибка. STATUS содержит информацию о типе ошибки.
STATUS	OUT	Word	Информация о состоянии включает в себя информацию об ошибке. (См. таблицу "Коды условий для ERROR и STATUS" ниже.)

Оба коммуникационных партнера выполняют инструкцию TCON, чтобы настроить и установить коммуникационное соединение. С помощью параметров задаются активная и пассивная конечные точки коммуникации. После настройки и установки соединения оно автоматически поддерживается и контролируется CPU.

Если соединение завершается, например, из-за разрыва линии или удаленным участником, то активный участник пытается восстановить его. Нет необходимости повторно выполнять инструкцию TCON.

При выполнении инструкции TDISCON или при переходе CPU в режим STOP текущее соединение завершается, а настроенное соединение удаляется. Для повторной настройки и установки соединения потребуется заново выполнить TCON.

Таблица 11-34 Коды условий ERROR иSTATUS для TCON и TDISCON

ERROR	STATUS	Описание
0	0000	Соединение было успешно установлено.
0	7000	Нет активных операций
0	7001	Запустить обработку операции, установить соединение (TCON) или завершить соединение (TDISCON)
0	7002	Следующий вызов (REQ не имеет значения), установить соединение (TCON) или завершить соединение (TDISCON)
1	8086	Параметр ID выходит за пределы допустимого диапазона адресов.
1	8087	TCON: Достигнуто максимальное число соединений; дальнейшие соединения невозможны.
1	809B	TCON: local_device_id в описании соединения не соответствует CPU.
1	80A1	TCON: Соединение или порт заняты пользователем.
1	80A2	TCON: Локальный или удаленный порт занят системой.
1	80A3	Попытка заново установить существующее соединение (TCON) или завершить несуществующее соединение (TDISCON).
1	80A4	TCON: Недействительный IP-адрес удаленной точки соединения, он соответствует локальному IP-адресу
1	80A5	TCON: ID соединения (Страница 832) уже используется.
1	80A7	TCON: Ошибка обмена данными: Инструкция TDISCON выполнена до завершения TCON. TDISCON сначала должна полностью завершить указанное ID соединение.
1	80B2	TCON: Параметр CONNECT указывает на блок данных, который был сгенерирован с атрибутом "Сохранять только в загружаемой памяти".

ERROR	STATUS	Описание
1	80B4	TCON: При установке пассивного соединения через ISO on TCP (connection_type = B#16#12), код состояния 80B4 предупреждает о том, что введенный TSAP не соответствовал одному из следующих требований к адресу: • Для длины локального TSAP равной 2 и значения идентификатора E0 или E1 (в шестнадцатеричном формате) для первого байта, второй байт должен быть 00 или 01. • Для длины локального TSAP равной 3 или больше и значения идентификатора E0 или E1 (в шестнадцатеричном формате) для первого байта, второй байт должен быть 00 или 01, а все остальные байты должны быть допустимыми ASCII-символами. • Для длины локального TSAP равной 3 или больше и первого байта значения идентификатора TSAP, не имеющего значения E0 или E1 (в шестнадцатеричном формате), все байты TSAP ID должны быть допустимыми ASCII-символами. Действительные ASCII-символы представляют собой байтовые значения от 20 до 7E (в шестнадцатеричном формате).
1	80B5	TCON: В типе соединения 13 = UDP допускается только пассивное установление соединения.
1	80B6	TCON: Ошибка параметризации в параметре CONNECTION_TYPE SDT TCON_Param.
1	80B7	TCON: Ошибка в одном из следующих параметров блока данных для описания соединения: • block_length • local_tsap_id_len • rem_subnet_id_len • rem_staddr_len • rem_tsap_id_len • next_staddr_len Примечание: При использовании TCON в пассивном TCP режиме, LOCAL_TSAP_ID_LEN должен быть "2", а REM_TSAP_ID_LEN - "0".
1	80B8	TCON: Параметры в локальном описании соединения и ID параметров различаются.
1	80C3	TCON: Все ресурсы соединения заняты.
1	80C4	Временная ошибка обмена данными: • В настоящий момент соединение не может быть установлено (TCON). • Сконфигурированное соединение удаляется TDISCON (TCON). • Указанное соединение устанавливается (TDISCON). • Интерфейс находится в процессе приема новых параметров (TCON и TDISCON).

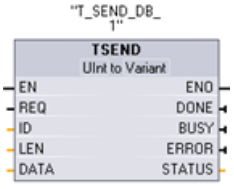
TSEND и TRCV

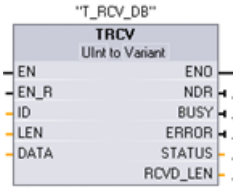
Примечание

При выполнении в открытых коммуникационных соединениях пользователя через PROFINET инструкции TSEND без соответствующей инструкции TRCV на удаленном устройстве, может случиться, что инструкция TSEND может оставаться неопределенно долго в состоянии "Busy", ожидая приема данных инструкцией TRCV. В этом состоянии выход "Busy" инструкции TSEND установлен и выход "Состояние" имеет значение "0x7002". Такое условие может возникнуть, если передается больше 4096 байт данных. Проблема устраняется при следующем выполнении инструкции TRCV.

Таблица 11-35 Инструкции TSEND и TRCV

LAD/FBD	SCL	Описание

	`"TSEND_DB"(`
	`req:=_bool_in_,`
	`ID:=_word_in_,`
	`len:=_udint_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`data:=_variant_inout_);`
		TCP и ISO on TCP: TSEND передает через коммуникационное соединение данные с CPU на станцию-партнер.

	`"TRCV_DB"(`
	`en_r:=_bool_in_,`
	`ID:=_word_in_,`
	`len:=_udint_in_,`
	`ndr=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`rcvd_len=>_udint_out_,`
	`data:=_variant_inout_);`
		TCP и ISO on TCP: TRCV принимает через коммуникационное соединение данные от станции-партнера на CPU.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Таблица 11-36 Типы данных для параметров TSEND и TRCV

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	TSEND: Запускает операцию передачи по положительному фронту. Данные передаются из указанной с помощью DATA и LEN области.
EN_R	IN	Bool	TRCV: Активирует CPU для приема, если EN_R = 1, тоTRCV готова к приему. Операция приема обрабатывается.

Параметр и тип данных		Тип данных	Описание
ID	IN	CONN_OUC (Word)	Это ссылка на соответствующее соединение. ID должен совпадать с соответствующим ID параметра в локальном описании соединения. Диапазон значений: от W#16#0001 до W#16#0FFF
LEN	IN	UInt	Максимальное количество байтов, которые должны быть отправлены (TSEND) или получены (TRCV): • По умолчанию = 0: Параметр DATA определяет длину передаваемых (TSEND) или принимаемых (TRCV) данных. • Ad hoc режим = 65535: Для приема данных установлена переменная длина (TRCV).
DATA	IN_OUT	Variant	Указатель на область данных для передачи (TSEND) или приема (TRCV). Область данных содержит адрес и длину. Адрес относится к одной из областей памяти I, Q или M или к DB.
DONE	OUT	Bool	TSEND: • 0: Операция еще не запущена или еще выполняется. • 1: Операция выполнена без ошибок.
NDR	OUT	Bool	TRCV: • NDR = 0: Операция еще не запущена или еще выполняется. • NDR = 1: Операция была успешно выполнена.
BUSY	OUT	Bool	• BUSY = 1: Операция еще не завершена. Новая операция не может быть запушена. • BUSY = 0: Операция завешена.
ERROR	OUT	Bool	ERROR = 1: При обработке возникла ошибка. STATUS содержит информацию о типе ошибки.
STATUS	OUT	Word	Информация о состоянии включает в себя информацию об ошибке. (См. таблицу "Коды условий для ERROR и STATUS" ниже.)
RCVD_LEN	OUT	Int	TRCV: Фактический объем принятых данных в байтах

Примечание

Для запуска операции передачи инструкции TSEND требуется переход 0 на 1 во входном параметре REQ. Параметр BUSY при обработке устанавливается на 1. Завершение операции передачи обозначается установкой одного из параметров DONE или ERROR на 1 на один цикл. В течение этого промежутка времени переход 0 на 1 во входном параметре REQ игнорируется.

Принцип работы TRCV

Инструкция TRCV записывает принятые данные в область приема, которая определяется следующими двумя переменными:

- Определяет начало области
- Длина области или значение, предоставляемое на входе LEN, если он не 0

Примечание

Настройка по умолчанию параметра LEN (LEN = 0) на основе параметра DATA определяет длину передаваемых данных. Рекомендуется, чтобы передаваемые инструкцией TSEND данные имели тот же размер, что и параметр DATA инструкции TRCV.

Если для параметра LEN используется настройка по умолчанию и данные должны передаваться партиями меньшего размера, чем параметр DATA, то действуют следующие правила. Рекомендуется оставить бит EN_R на 1, пока соответствующая инструкция TSEND передаст соответствующий объем данных, чтобы заполнить параметр DATA из TRCV. Если размер передаваемых TSEND данных не соответствует размеру параметра DATA из TRCV, то TRCV остается в состоянии "Занято" (код состояния: 7002) до тех пор, пока бит EN_R = 1, пока общий размер передаваемых TSEND данных не сравняется с параметром DATA из TRCV. Если бит EN_R из TRCV принимает импульсы, то он должен получить столько импульсов, сколько раз должна быть выполнена инструкция TSEND для приема данных.

Буфер параметра DATA из TRCV показывает новые полученные данные только после того, как размер данных сравняется с размером буфера параметра DATA.

Как только все данные операции получены, они передаются из TRCV в область приема, и NDR устанавливается на 1.

Таблица 11-37 Запись данных в область приема

Вариант протокола	Запись данных в область приема	Параметр "connection_type"	Значение параметра LEN	Значение параметра RCVD_LEN (в байтах)
TCP	Ad hoc режим	B#16#11	65535	От 1 до 1472
TCP	Получение данных с указанной длиной	B#16#11	0 (рекомендуется) или от 1 до 8192, кроме 65535	От 1 до 8192
ISO on TCP	Ad hoc режим	B#16#12	65535	От 1 до 1472
ISO on TCP	управляется протоколом	B#16#12	0 (рекомендуется) или от 1 до 8192, кроме 65535	От 1 до 8192

Примечание

Ad hoc режим

Ad hoc режим доступен только для вариантов протокола TCP и ISO on TCP. Ad hoc режим можно установить, присвоив параметру LEN значение "65535". Область приема идентична указанной параметром DATA области. Длина принятых данных выводится в параметре RCVD_LEN. Непосредственно после получения блока данных инструкция TRCV записывает данные в область приема и устанавливает NDR на "1".

Если данные хранятся в "оптимизированном" DB (только символьная адресация), то данные могут быть приняты только в массивах данных типа Byte, Char, USInt иSInt.

Примечание

Импорт STEP 7 проектов S7-300/400 с Ad hoc режимом в S7-1200

В STEP 7 проектах S7-300/400 Ad hoc режим устанавливается путем присвоения параметру LEN значения "0". Для S7-1200 можно установить Ad hoc режим, присвоив параметру LEN значение "65535".

При импорте STEP 7 проекта S7-300/400, включающего в себя Ad hoc режим, в S7-1200, необходимо изменить параметр LEN на 65535.

Коды условий TSEND и TRCV Error и Status

ERROR	STATUS	Описание
0	0000	• Операция передачи была выполнена без ошибок (TSEND) • Новые данные приняты: Фактическая длина полученных данных отображается в RCVD_LEN (TRCV).
0	7000	• Нет активных операций (TSEND) • Блок не готов к получению (TRCV)
0	7001	• Запустить обработку операции, идет отправка данных: Во время обработки операционная система обращается к данным в области передачиDATA (TSEND). • Блок готов к приему, операция приема была активирована (TRCV).
0	7002	• Следующий вызов (REQ не имеет значения), операция находится в обработке: Во время обработки операционная система обращается к данным в области передачиDATA (TSEND). • Следующий вызов, операция приема обрабатывается: Во время обработки данные записываются в область приема. Поэтому возникновение ошибки может привести появлению противоречивых данных в области приема (TRCV).
1	8085	• Параметр LEN больше, чем максимально допустимое значение (TSEND) и (TRCV). • Параметр LEN или DATA изменился после последнего выполнения инструкции (TRCV).
1	8086	Параметр ID за пределами допустимой области адресов.
1	8088	Параметр LEN больше, чем указанная в DATA область памяти.

ERROR	STATUS	Описание
1	80A1	Ошибка обмена данными: • Указанное соединение еще не установлено (TSEND и TRCV). • Указанное соединение завершается. Операция передачи или приема по этому соединению невозможна (TSEND и TRCV). • Интерфейс инициализируется заново (TSEND). • Интерфейс находится в процессе приема новых параметров (TRCV).
1	80C3	Внутренний дефицит ресурсов: Блок с таким ID уже обрабатывается в другом классе приоритета.
1	80C4	Временная ошибка обмена данными: • В настоящий момент соединение с партнером не может быть установлено • Интерфейс находится в процессе приема новых настроек для параметров, либо в данный момент выполняется установка соединения.

Протоколы соединения Ethernet

У каждого CPU есть встроенный порт PROFINET с поддержкой стандартной PROFINET коммуникации. Инструкции TSEND_C, TRCV_C, TSEND и TRCV поддерживают TCP и ISO on TCP протоколы Ethernet.

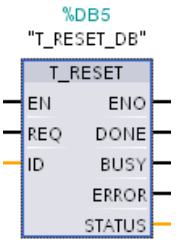
Дополнительную информацию см. в "Конфигурация устройства: Конфигурирование пути соединения между локальным и партнерским CPU (Страница 812)".

11.2.8.12 Инструкция T_RESET (разорвать и восстановить существующее соединение)

Инструкция "T_RESET" разрывает и затем восстанавливает существующее соединение:

Таблица 11-38 Инструкция T_RESET

LAD/FBD	SCL	Описание

	`"T_RESET_DB"(`
	`req:=_bool_in_,`
	`id:=_word_in_,`
	`done=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_);`
		Инструкция "T_RESET" разрывает и затем восстанавливает существующее соединение.

Локальные конечные точки соединения при этом сохраняются. Он создаются автоматически:

- Если соединение было сконфигурировано и загружено в CPU
- Если соединение было создано с помощью пользовательской программы, например, посредством вызова инструкции "TCON (Страница 864)".

Инструкция "T_RESET" выполняется для всех видов соединений, независимо от того, был ли использован для соединения локальный интерфейс CPU или интерфейс CM/CP.

Исключением являются соединения для передачи данных в Ad-hoc режиме с TCP, так как на них невозможно ссылаться через ID соединения.

Если инструкция "T_RESET" была вызвана с параметром REQ , то указанное параметром ID соединение разрывается и, если необходимо, буфер приема и передачи для данных очищается. Разрыв соединения означает и завершение текущей передачи данных. Поэтому возможна потеря данных, если они передаются в настоящий момент. После определенный в качестве активного участника соединения CPU автоматически пытается восстановить прерванное коммуникационное соединение. Поэтому вызов инструкции "TCON (Страница 864)" для восстановления коммуникационного соединения не нужен.

Выходные параметры DONE, BUSY иSTATUS показывают состояние операции.

Типы данных для параметров

В приведенной ниже таблице показаны параметры инструкции T_RESET:

Параметр	Объявление	Тип данных	Область памяти	Описание
REQ	Input	BOOL	I, Q, M, L, D, T, C или постоянная	Управляющий параметр REQUEST запускает задание по разрыву заданного по ID соединения. Операция начинается по положительному фронту.
ID	Input	CONN_OUC (WORD)	L, D или постоянная	Ссылка на соединение с пассивным участником, которое должно быть завершено. ID должен совпадать с соответствующим ID параметра в локальном описании соединения. Диапазон значений: W#16#0001 до W#16#0FFF
DONE	Output	BOOL	I, Q, M, D, L	Параметр состояния DONE • 0: Операция еще не запущена или еще выполняется. • 1: Операция выполнена без ошибок.
BUSY	Output	BOOL	I, Q, M, D, L	Параметр состояния BUSY • 0: Операция завешена. • 1: Операция еще не завершена.
ERROR	Output	BOOL	I, Q, M, D, L	Параметр состояния ERROR • 0: Ошибки отсутствуют. • 1: Ошибки при обработке. Параметр STATUS содержит подробную информацию о типе ошибки
STATUS	Output	WORD	I, Q, M, D, L	Параметр состояния STATUS Информация об ошибке (см. таблицу "Параметр STATUS").

Параметр STATUS

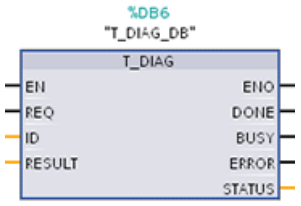
Бит ошибки	STATUS* (W#16#...)	Описание
0	0000	Ошибки отсутствуют.
0	0001	Соединение не установлено.
0	7001	Запущен разрыв соединения.
0	7002	Соединение завершается.
1	8081	В параметре ID указано неизвестное соединение.

11.2.8.13 Инструкция T_DIAG (проверить состояние соединения и считать информацию)

Инструкция T_DIAG проверяет состояние соединения и считывает дополнительную информацию на локальной конечной точке этого соединения:

Таблица 11-39 Инструкция T_DIAG

LAD/FBD	SCL	Описание

	`"T_DIAG_DB"(`
	`req:=_bool_in_,`
	`id:=_word_in_,`
	`done=>_bool_out_,`
	`error=>_bool_out_,`
	
	`status=>_dword_out_);`
		Инструкция T_DIAG проверяет состояние соединения и считывает дополнительную информацию на локальной конечной точке этого соединения.

Инструкция T_DIAG работает следующим образом:

- Соединение задается параметром ID. Могут быть считаны обе сконфигурированные в редакторе соединений точки соединения и запрограммированные точки соединения (напр., через инструкцию TCON).
  Диагностика временных конечных точек соединения (напр., созданных при соединении с инженерной станцией конечных точек) невозможна, так как при этом процессе не создается ID соединения.
- Считанная информация соединения сохраняется в одну из указанных в параметре RESULT структур.
- Выходной параметр STATUS указывает, было ли возможно считывание информации о соединении. Информация о соединении в структуре в параметре RESULT является достоверной только в том случае, если инструкция T_DIAG была завершена с STATUS = W#16#0000 и ERROR = FALSE.
  В случае ошибки информация о соединении не анализируется.

Возможная информация о соединении

Структура TDiag_Status может использоваться для чтения информации о соединении в параметре RESULT. Структура TDiag_Status включает в себя только важнейшие сведения о конечной точке соединения, такие как, используемый протокол, состояние соединения и количество отправленных или полученных байт данных.

Структура и параметры TDiag_Status описываются ниже (см. таблицу для структуры TDIAG_Status).

Типы данных для параметров

В приведенной ниже таблице показаны параметры инструкции T_DIAG:

Параметр	Объявление	Тип данных	Область памяти	Описание
REQ	Input	BOOL	I, Q, M, L, D, T, C или константа	Запускает инструкцию, чтобы проверить соединение, определенное в параметре ID, когда есть положительный фронт.
ID	Input	CONN_OUC (WORD)	L, D или постоянная	Ссылка на назначенное соединение. Диапазон значений: W#16#0001 до W#16#0FFF
RESULT	InOut	VARIANT	D	Указатель на структуру, в которой сохраняется информация о соединении. Структура TDiag_Status может использоваться в параметре RESULT (описание можно найти в таблице для структуры TDIAG_Status).
DONE	Output	BOOL	I, Q, M, D, L	Параметр состояния: • 0: Инструкция еще не запущена либо еще находится в обработке. • 1: Инструкция была выполнена без ошибок.
BUSY	Output	BOOL	I, Q, M, D, L	Параметр состояния: • 0: Инструкция еще не запущена либо уже завершена. • 1: Инструкция еще не завершена. Новая операция не может быть запушена.
ERROR	Output	BOOL	I, Q, M, D, L	Параметр состояния: • 0: Ошибки отсутствуют. • 1: Возникла ошибка.
STATUS	Output	WORD	I, Q, M, D, L	Состояние инструкции

Параметры BUSY, DONE и ERROR

Состояние выполнения T_DIAG можно проверить с помощью параметров BUSY, DONE, ERROR и STATUS. Параметр BUSY показывает состояние обработки. С помощью параметра DONE можно проверить, успешно ли была выполнена инструкция. Параметр ERROR устанавливается, если во время выполнения инструкции T_DIAG возникли ошибки.

В следующей таблице представлены отношения между параметрами BUSY, DONE и ERROR:

BUSY	DONE	ERROR	Описание
1	-	-	Инструкция обрабатывается.
0	1	0	Инструкция выполнена успешно. Данные в указанной через RESULT структуре являются достоверными только в этом случае.

BUSY	DONE	ERROR	Описание
0	0	1	Инструкция завершена с ошибкой. Причина ошибки выводится в параметре STATUS.
0	0	0	Новая инструкция не была назначена.

Параметр STATUS

В следующей таблице показан смысл значений параметра STATUS:

Бит ошибки	STATUS* (W#16#...)	Описание
0	0000	Инструкция T_DIAG выполнена успешно. Данные в структуре, указанной в параметре RESULT, могут быть проанализированы.
0	7000	Обработка инструкции неактивна.
0	7001	Обработка инструкции запущена.
0	7002	Считывается информация о соединении (параметр REQ не имеет значения).
1	8086	Значение в параметре ID находится вне допустимого диапазона (W#16#0001 до W#16#0FFF).
1	8089	Параметр RESULT указывает на недопустимый тип данных (только структуры TDIAG_Status и TDIAG_StatusExt).
1	80A3	Параметр ID ссылается на конечную точку соединения, которая не существует. Эта ошибка возникает при программируемом соединении и после вызова инструкции TDISCON.
1	80C4	Внутренняя ошибка. Доступ к конечной точке соединения временно невозможен.

Структура TDIAG_Status

В следующей таблице содержаться подробности о структуре TDIAG_Status. Значение каждого элемента действительно только, если инструкция была выполнена без ошибок. Если произойдет ошибка, то содержимое параметров не изменится:

Название	Тип данных	Описание
Следующие параметры находятся в структуре TDIAG_Status:		
InterfaceID	HW_ANY	ID интерфейса (LADDR) CPU или CM/CP.
ID	CONN_OUC	ID диагностируемого соединения. После успешного вызова значение этого элемента идентично ID параметра инструкции T_DIAG.

Название	Тип данных	Описание
ConnectionType	BYTE	Тип протокола, используемый для соединения: • 0x01: Не используется. • ... • 0x0B: Протокол TCP (IP_v4) • 0x0C: Протокол ISO-on-TCP (RFC1006) • 0x0D: Протокол TCP (DNS) • 0x0E: Dial-in протокол • 0x0F: Протокол WDC • 0x10: Протокол SMTP • 0x11: Протокол TCP • 0x12: TCP- и ISO-on-TCP протокол (RFC1006) • 0x13: Протокол UDP • 0x14: Зарезервировано • 0x15: Протокол доступа к шине PROFIBUS (FDL) • 0x16: Транспортный протокол по ISO 8073 (нативный ISO) • ... • 0x20: SMTP- или SMTPS-протокол - на базе IPv4 • 0x21: SMTP- или SMTPS-протокол - на базе IPv6 • 0x22: SMTP- или SMTPS-протокол - на базе FQDN (**F**ully **Q**ualified **D**omain **N**ame) • ... • 0x70: S7-соединение • Другие: Зарезервировано
ActiveEstablished	BOOL	• FALSE: Локально, пассивная конечная точка соединения • TRUE: Локально, активная конечная точка соединения
State	BYTE	Текущий статус конечной точки соединения • 0x00: Не используется. • 0x01: Соединение разорвано. Временное состояние, например, после вызова инструкции T_RESET. Затем система автоматически пытается восстановить соединение. • 0x02: Активная конечная точка пытается установить соединение с удаленным коммуникационным партнером. • 0x03: Пассивная конечная точка соединения ожидает установления соединения с удаленным коммуникационным партнером. • 0x04: Соединение установлено. • 0x05: Соединение завершается. Это может происходить вследствие вызова инструкции "T_RESET" или "T_DISCON". Другие возможные причины - ошибки протокола и разрывы линии. • 0x06..0xFF: Не используется.

Название	Тип данных	Описание
Kind	BYTE	Режим конечной точки соединения: • 0x00: Не используется. • 0x01: Настроенное статическое соединение, которое было сконфигурировано и загружено в CPU. • 0x02: Настроенное динамическое соединение, которое было сконфигурировано и загружено в CPU (в настоящее время не поддерживается). • 0x03: Запрограммированное соединение, созданное в пользовательской программе с помощью инструкции "TCON". Вызов инструкции TDISCON или переход CPU в состояние STOP нарушил конечную точку соединения. • 0x04: Временное, динамическое соединение, установленное, например, инженерной станцией (ES) или станцией оператора (OS). (Этот тип соединения не может в настоящее время быть диагностирован, так как нет ID). • 0x05..0xFF: Не используется.
SentBytes	UDINT	Число переданных байтов данных.
ReceivedBytes	UDINT	Число принятых байтов данных.

11.2.8.14 Инструкция TMAIL_C (передача Email через Ethernet интерфейс CPU)

Обзор

С помощью инструкции "TMAIL_C" можно отправить E-Mail через Ethernet интерфейс S7-1200 CPU.

Инструкция TMAIL_C имеет две функции:

- E-Mail через CPU интерфейс (только SMTP без SSL)
- E-Mail через CP интерфейс (либо SMTP без SSL, либо SMTP с SSL). Если требуется функциональность SSL, то следует установить входной параметр CERTINDEX изTMAIL_C = 1 и использовать интерфейс CP. Кроме этого, в хранилище сертификатов CP должен находиться нужный сертификат.

Если используется тип системных данных (SDT) TMail_V4, то письма могут отправляться только в незащищенном режиме через встроенный Ethernet порт S7-1200 CPU. SDT TMail_V6 не поддерживается для отправки писем через встроенный Ethernet порт S7-1200 CPU.

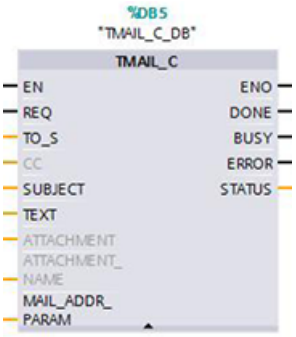
Если используется SDT TMail_FQDN, то письма могут отправляться только через коммуникационный процессор (CP). Нельзя использовать SDT TMail_FQDN для писем через встроенный Ethernet порт S7-1200 CPU.

От TMAIL_C версии V6.0 или выше и прошивки S7-1200 CPU от версии V4.4 можно использовать инструкцию TMAIL_C для передачи письма по защищенному каналу через встроенный Ethernet порт S7-1200 CPU. Необходимые для процесса передачи данные определяются в параметре MAIL_ADDR_PARAM с помощью SDT TMail_V4_SEC или TMail_QDN_SEC.

Инструкция TMAIL_C может использоваться только в том случае, если оборудование сконфигурировано и если сетевая инфраструктура обеспечивает коммуникационное соединение с почтовым сервером.

Таблица 11-40 Инструкция TMAIL_C

LAD/FBD	SCL	Описание

	`"TMAIL_C_DB"(`
	`req:=_bool_in_,`
	`to_s:=_string_in_,`
	`cc:=_string_in_,`
	`subject:=_string_in_,`
	`text:=_string_in_,`
	`attachment:=_variant_in_,`
	`attachment_name:=_string_in_,`
	`mail_addr_param:=_string_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_);`
		С помощью инструкции TMAIL_C можно отправить E-Mail через Ethernet интерфейс S7-1200 CPU.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Содержание письма и данные соединения определяются с помощью следующих параметров:

- С помощью параметров TO_S и CC определяются адреса получателей.
- Содержание письма определяется с помощью параметров SUBJECT и TEXT.
- С помощью указателей VARIANT в параметрах ATTACHMENT и ATTACHMENT_NAME можно определить вложение.
- Инструкция TMAIL_C определяет данные соединения и выполняет адресацию и аутентификацию для почтового сервера в параметре MAIL_ADDR_PARAM через следующие SDT:
  - TMail_V4
  - TMail_V4_SEC
  - TMail_QDN_SEC

  Для интерфейса S7-1200 CPU с версией прошивки V4.3 надо использовать SDT TMail_V4. В этом случае передача писем возможна только по SMTP.
- Передача письма запускается при переключении фронта для параметра REQ с "0" на "1".
- Инструкция TMAIL_C сообщает о состоянии операции на основе выходных параметров "BUSY", "DONE", "ERROR" и "STATUS".

Нельзя отправить SMS напрямую с помощью инструкции TMAIL_C. Может ли письмо быть перенаправлено почтовым сервером в качестве SMS, зависит от телекоммуникационного провайдера.

Принцип работы инструкции

Инструкция TMAIL_C работает асинхронно, т.е. ее выполнение растягивается на несколько вызовов. При вызове инструкции "TMAIL_C" необходимо указать экземпляр.

Соединение с почтовым сервером прерывается в следующих случаях:

- Если CPU переходит в состояние STOP при активной TMAIL_C.
- При наличии проблем со связью на шине Industrial Ethernet. В таких случаях отправка электронной почты прерывается, и она не доходит до получателя.

Соединение завершается и после успешного выполнения инструкции и отправки письма.

ВНИМАНИЕ
Изменение программ пользователя Можно изменять те части своей программы пользователя, которые непосредственно затрагиваю вызовы TMAIL_C только при выполнении следующих условий: • CPU находится в состояние STOP • В настоящий момент не выполняется отправка почты (т.е. REQ = 0 и BUSY = 0). В первую очередь это касается удаления и замены программных блоков, содержащих вызовы TMAIL_C или экземпляра TMAIL_C. Игнорирование этого ограничения может заблокировать коммуникационные ресурсы. Система автоматизации может перейти в неопределенное состояние при использовании функций коммуникации TPC/IP через Industrial Ethernet. После передачи изменений необходимо выполнить горячий или холодный пуск CPU.

Целостность данных

Параметры TO_S, CC, SUBJECT, TEXT, ATTACHMENT и MAIL_ADDR_PARAM применяются инструкцией TMAIL_C во время ее работы, то есть они могут быть изменены только после того, как операция была завершена (BUSY = 0).

SMTP аутентификация

Под аутентификацией здесь понимается процедура проверки подлинности, например, с помощью запроса пароля.

При использовании интерфейса S7-1200 CPU, инструкция TMAIL_C поддерживает процедуру SMTP аутентификации AUTH-LOGIN, которая используется большинством почтовых серверов. Для получения информации о процедуре аутентификации для

почтового сервера, смотри руководство по почтовому серверу или веб-сайт Интернет-провайдера.

- Для использования метода аутентификации AUTH-LOGIN в инструкции TMAIL_C требуется имя пользователя, с которым он может получить доступ к почтовому серверу. Это имя соответствует пользователю, для которого создавалась почтовая учетная запись на почтовом сервере. С помощью параметра UserName оно передается в структуру в параметре MAIL_ADDR_PARAM.
  Если имя пользователя не указано в параметре MAIL_ADDR_PARAM, то метод аутентификации AUTH-LOGIN не применяется. В таком случае электронная почта будет отправляться без аутентификации.
- Чтобы войти в систему, инструкция TMAIL_C требует также соответствующего пароля. Он соответствует паролю, который был определен при создании почтовой учетной записи. С помощью параметра PassWord он передается в структуру в параметре MAIL_ADDR_PARAM.

Типы данных для параметров

В приведенной ниже таблице показаны параметры инструкции TMAIL_C:

Параметр	Объявление	Тип данных	Область памяти	Описание
REQ	Input	BOOL	I, Q, M, L, D, T, C или постоянная	Управляющий параметр REQUEST: Активирует отправку электронной почты при положительном фронте.
TO_S (Страница 904)	Input	STRING	D	Адреса получателей STRING с максимальной длиной 180 символов (байт). Для формата адреса электронной почты см. пример в описании параметра.
CC (Страница 904)	Input	STRING	D	Адреса получателей копии (опционально) STRING с максимальной длиной 180 символов (байт). Тот же формат адреса электронной почты, что и в параметре TO_S. Если здесь назначена пустая строка, копии электронного письма не посылаются.
SUBJECT	Input	STRING	D	Тема электронного письма STRING с максимальной длиной 180 символов (байт).
TEXT	Input	STRING	D	Текст электронного письма (опционально) STRING с максимальной длиной 180 символов (байт). Если здесь назначена пустая строка, то электронное письмо посылается без текста.

Параметр	Объявление	Тип данных	Область памяти	Описание
ATTACHMENT	Input	VARIANT	D	Почтовое вложение (опционально) Ссылка на массив (ArrayOfChar, ArrayOfByte, ArrayOfWord, ArrayOfDWord, or String) с максимальной длиной в 64 байт. Примечание: Если значение в параметре ATTACHMENT не назначено или назначена пустая строка, то электронное письмо посылается без вложения.
ATTACHMENT_NAME	Input	VARIANT	D	Наименование почтового вложения (опционально) Ссылка на строку символов с максимальной длиной 50 символов (байт) для определения имени файла вложения. Если этому параметру назначается пустая символьная строка, то почтовое вложение оправляется с именем файла "attachment.bin". С помощью параметра AttachmentName можно присвоить вложению имя, которое будет отображаться при получении письма участником процесса коммуникации. SDT TMail_FileReference автоматически использует параметр FileName для параметра AttachmentName. Если используется SDT TMail_FileReference, то параметр AttachmentName не имеет значения. Он может не заполняться. Если при использовании SDT TMail_FileReference ввести параметр AttachmentName, то инструкция TMAIL_C создает ошибку. Дополнительную информацию можно найти здесь: "Коды ошибок, параметр SFB_STATUS блока данных экземпляра".
MAIL_ADDR_PARAM (Страница 896)	Input	VARIANT	D	Параметры соединения и адрес почтового сервера Параметры соединения определяются в SDT со структурой TMail_V4, TMail_FQDN, TMail_V4_SEC или TMail_QDN_SEC (см. Описание параметров).

Параметр	Объявление	Тип данных	Область памяти	Описание
DONE (Страница 904)	Output	BOOL	I, Q, M, D, L	Параметр состояния • DONE = 0: Операция еще не запущена или еще выполняется. • DONE = 1: Операция была выполнена без ошибок.
BUSY (Страница 904)	Output	BOOL	I, Q, M, D, L	Параметр состояния • BUSY = 0: Обработка TMAIL_C была остановлена. • BUSY = 1: Передача письма еще не завершена.
ERROR (Страница 904)	Output	BOOL	I, Q, M, D, L	Параметр состояния • ERROR = 0: Ошибок не обнаружено. • ERROR = 1: При обработке возникла ошибка. STATUS содержит информацию о типе ошибки.
STATUS (Страница 908)	Output	WORD	I, Q, M, D, L	Параметр состояния Возвращаемое значение или информация об ошибке инструкции TMAIL_C (см. Описание параметра).

Примечание

Опциональные параметры

Опциональные параметры CC, TEXT и ATTACHMENT передаются вместе с письмом только в том случае, если соответствующие параметры имеют строку длиной > 0.

Параметр MAIL_ADDR_PARAM

В параметре MAIL_ADDR_PARAM определяется соединение для передачи почты в структуре TMail_v4, TMail_FQDN, TMail_V4_SEC или TMail_QDN_SEC и сохраняется адрес почтового сервера и детали для входа.

Тип системных данных (SDT), используемый в параметре MAIL_ADDR_PARAM, зависит от формата адресации для почтового сервера:

- TMail_V4: Адресация через IP-адрес (IPv4)
- TMail_FQDN: Адресация через полное доменное имя (FQDN)
- TMail_V4_SEC: Безопасная адресация через IP-адрес (IPv4)
- TMail_QDN_SEC: Безопасная адресация через полное доменное имя (FQDN)

Какой SDT может использоваться, зависит от интерфейса, адресуемого в параметре InterfaceId. Если необходимо использовать инструкцию TMAIL_C с внутренним интерфейсом CPU, то в параметре MAIL_ADDR_PARAM следует использовать SDT TMail_V4, TMail_V4_SEC или TMail_QDN_SEC.

Если используется SDT TMail_FQDN, то письма могут отправляться только через коммуникационный процессор (CP). Нельзя использовать SDT TMail_FQDN для писем через встроенный Ethernet порт S7-1200 CPU.

Таблица 11-41 **TMail_V4: адресация почтового сервера через IP-адрес (IPv4)**

Параметр	Тип данных	Описание
TMail_v4	Struct	
InterfaceId	LADDR	Аппаратный идентификатор интерфейса Ethernet
ID	CONN_OUC	Идентификатор соединения
ConnectionType	BYTE	Тип соединения. Выбрать для IPv4 в качестве типа соединения 16#20.
ActiveEstablished	BOOL	Активное/пассивное установление соединения. CPU это всегда SMTP клиент.
CertIndex	BYTE	= 0: Используется SMTP (**S**imple **M**ail **T**ransfer **P**rotocol). SMTP должен использоваться, если электронное письмо посылается через интерфейс S7-1200 CPU.
WatchDogTime	TIME	Контроль времени выполнения. С помощью этого параметра можно определить макс. время выполнения для передачи. Это значение определяет, как долго может выполняться инструкция TMAIL_C, прежде чем произойдет превышение времени и выполнение инструкции TMAIL_C будет остановлено. От версии 6.0 инструкции TMAIL_C разрешено значение WatchDogTime = 0, означающее выключение таймера инструкции TMAIL_C при выполнении. Но можно ввести ненулевое значение для WatchDogTime, чтобы сделать выполнение инструкции TMAIL_C более детерминированным. Примечание: При медленном соединении установка соединения может занимать длительное время (около одной минуты). Поэтому при спецификации параметра WatchDogTime необходимо учитывать время на установление соединения.
MailServerAddress	IP_v4	IP-адрес почтового сервера. Согласно IPv4 в формате XXX.XXX.XXX.XXX (десятичный). Пример: 192.142.131.237
UserName	STRING[254]	Имя для входа на почтовый сервер
PassWord	STRING[254]	Пароль для входа на почтовый сервер
From	EMAIL_ADDR	Почтовый адрес отправителя, который определяется, используя следующие два параметра типа STRING. Пример: "myname@mymailserver.com".
LocalPartPlusAtSign	STRING[64]	Локальная часть адреса отправителя, включая символ "@". Пример: "myname@".
FullQualifiedDomainName	STRING[254]	Fully Qualified Domain Name (сокращенно FQDN) почтового сервера. Пример: "mymailserver.com".

Таблица 11-42 **TMail_FQDN *: Адресация почтового сервера с помощьюFQDN**

Параметр	Тип данных	Описание
TMail_FQDN	Struct	
InterfaceId	LADDR	Аппаратный идентификатор интерфейса Ethernet
ID	CONN_OUC	Идентификатор соединения
ConnectionType	BYTE	Тип соединения. Выбрать для FQDN в качестве типа соединения 16#22.
ActiveEstablished	BOOL	Активное/пассивное установление соединения. Коммуникационный процессор (CP) это всегда SMTP-клиент.
CertIndex	BYTE	= 0: Используется SMTP (**S**imple **M**ail **T**ransfer **P**rotocol).
WatchDogTime	TIME	Контроль времени выполнения. С помощью этого параметра можно определить макс. время выполнения для передачи. Примечание: При медленном соединении установка соединения может занимать длительное время (около одной минуты). Поэтому при спецификации параметра WatchDogTime необходимо учитывать время на установление соединения.
MailServerAddress	STRING[254]	FQDN (**F**ully **Q**ualified **D**omain **N**ame, полное доменное имя) почтового сервера. Адресация почтового сервера через FQDN. Пример: "www.mymailserver.com."
UserName	STRING[254]	Имя для входа на почтовый сервер
PassWord	STRING[254]	Пароль для входа на почтовый сервер
From	Struct	Почтовый адрес отправителя, который определяется, используя следующие два параметра типа STRING. Пример: "myname@mymailserver.com".
LocalPartPlusAtSign	STRING[64]	Локальная часть адреса отправителя, включая символ "@". Пример: "myname@".
FullQualifiedDomainName	STRING[254]	Полное доменное имя (сокращенно FQDN) почтового сервера. Пример: "mymailserver.com".

* Если используется SDT TMail_FQDN, то письма могут отправляться только через коммуникационный процессор (CP). Нельзя использовать SDT TMail_FQDN для писем через встроенный Ethernet порт S7-1200 CPU.

Таблица 11-43 **TMail_V4_SEC: адресация почтового сервера через IP-адрес (IPv4)**

Параметр	Тип данных	Описание
TMail_V4_SEC	Struct	

Параметр	Тип данных	Описание
InterfaceId	LADDR	Аппаратный идентификатор диапазона значений для Ethernet интерфейса: • 0 (новый): Операционная система самостоятельно выбирает подходящий встроенный порт. • Аппаратный идентификатор встроенного порта для использования.
ID	CONN_OUC	Ссылка на соединение: Диапазон значений: • 0 (новый): Операционная система выбирает свободный ID соединения из внутренней области. • 1 до 4095: Используемый ID соединения
ConnectionType	BYTE	Тип соединения. Выбрать для IPv4 в качестве типа соединения 16#20.
ActiveEstablishment	BOOL	Активное/пассивное установление соединения. CPU это всегда SMTP клиент.
WatchDogTime	TIME	Превышение времени при выполнении. С помощью этого параметра можно определить макс. время выполнения для передачи. Это значение определяет, как долго может выполняться инструкция TMAIL_C, прежде чем произойдет превышение времени и выполнение инструкции TMAIL_C будет остановлено. От версии 6.0 инструкции TMAIL_C разрешено значение WatchDogTime = 0, означающее выключение таймера инструкции TMAIL_C при выполнении. Но можно ввести ненулевое значение для WatchDogTime, чтобы сделать выполнение инструкции TMAIL_C более детерминированным. Примечание: При медленном соединении установка соединения может занимать длительное время (около одной минуты). Поэтому при спецификации параметра WatchDogTime необходимо учитывать время на установление соединения.
MailServerAddress	IP_V4	IP-адрес почтового сервера в формате IPv4: XXX.XXX.XXX.XXX (дес.) Пример: 192.142.131.237
UserName	STRING[254]	Имя пользователя Следует использовать свое "имя пользователя" для доступа к своему почтовому ящику, чтобы идентифицировать себя как владельца почтового ящика у провайдера электронной почты.
PassWord	STRING[254]	Пароль пользователя Следует использовать свой "пароль" для доступа к своему почтовому ящику, чтобы идентифицировать себя как владельца почтового ящика у провайдера электронной почты.
From	EMAIL_ADDR	Почтовый адрес отправителя, который определяется, используя следующие два параметра типа STRING. Пример: "myname@mymailserver.com"

Параметр		Тип данных	Описание
	LocalPartPlusAtSign	STRING[64]	Локальная часть адреса отправителя, включая символ "@". Пример: "myname@"
	FullQualifiedDomainName	STRING[254]	Полное доменное имя (сокращенно FQDN) почтового сервера. Пример: "mymailserver.com"
RemotePort		UINT	TCP порт почтового сервера
ActivateSecureConn		BOOL	• 1: Безопасное SMTP соединение • 0: SMTP соединение (незащищенное). В этом случае следующие параметры не имеют значения.
ExtTLSCapabilities		BYTE	В настоящее время не используется.
TLSServerCertRef		UDINT	Ссылка на сертификат X.509 V3 (CA) почтового сервера, с помощью которого TLS клиент проверяет аутентификацию TLS-сервера

Таблица 11-44 **TMail_QDN_SEC: Адресация почтового сервера с помощьюFQDN**

Параметр	Тип данных	Описание
TMail_QDN_SEC	Struct	

Параметр	Тип данных	Описание
InterfaceId	LADDR	Аппаратный идентификатор диапазона значений для Ethernet интерфейса: • 0 (новый): Операционная система самостоятельно выбирает подходящий встроенный порт. • Аппаратный идентификатор встроенного порта для использования.
ID	CONN_OUC	Ссылка на соединение: Диапазон значений: • 0 (новый): Операционная система выбирает свободный идентификатор соединения из внутренней области. • 1 до 4095: Используемый идентификатор соединения
ConnectionType	BYTE	Тип соединения. Выбрать для FQDN в качестве типа соединения 16#22.
ActiveEstablishment	BOOL	Активное/пассивное установление соединения. CPU это всегда SMTP клиент.
WatchDogTime	TIME	Превышение времени при выполнении. С помощью этого параметра можно определить макс. время выполнения для передачи. Это значение определяет, как долго может выполняться инструкция TMAIL_C, прежде чем произойдет превышение времени и выполнение инструкции TMAIL_C будет остановлено. От версии 6.0 инструкции TMAIL_C разрешено значение WatchDogTime = 0, означающее выключение таймера инструкции TMAIL_C при выполнении. Но можно ввести ненулевое значение для WatchDogTime, чтобы сделать выполнение инструкции TMAIL_C более детерминированным. Примечание: При медленном соединении установка соединения может занимать длительное время (около одной минуты). Поэтому при спецификации параметра WatchDogTime необходимо учитывать время на установление соединения.
MailServerQDN	STRING[254]	FQDN (**F**ully **Q**ualified **D**omain **N**ame, полное доменное имя) почтового сервера. Адресация почтового сервера выполняется через FQDN, которое должно оканчиваться на ".". Пример: "www.mymailserver.com."
UserName	STRING[254]	Имя пользователя Следует использовать свое "имя пользователя" для доступа к своему почтовому ящику, чтобы идентифицировать себя как владельца почтового ящика у провайдера электронной почты.
PassWord	STRING[254]	Пароль пользователя Следует использовать свой "пароль" для доступа к своему почтовому ящику, чтобы идентифицировать себя как владельца почтового ящика у провайдера электронной почты.
From	EMAIL_ADDR	Почтовый адрес отправителя, который определяется, используя следующие два параметра типа STRING. Пример: "myname@mymailserver.com"

Параметр		Тип данных	Описание
	LocalPartPlusAtSign	STRING[64]	Локальная часть адреса отправителя, включая символ "@". Пример: "myname@"
	FullQualifiedDomainName	STRING[254]	Полное доменное имя (сокращенно FQDN) почтового сервера. Пример: "mymailserver.com"
RemotePort		UINT	TCP порт почтового сервера
ActivateSecureConn		BOOL	• 1: Безопасное SMTP соединение • 0: SMTP соединение (незащищенное). В этом случае следующие параметры не имеют значения.
ExtTLSCapabilities		BYTE	В настоящее время не используется.
TLSServerCertRef		UDINT	Ссылка на сертификат X.509 V3 (CA) почтового сервера, с помощью которого TLS клиент проверяет аутентификацию TLS-сервера

Параметры TO_S и CC

Для инструкций TMAIL_C до версии 6.0 и S7-1200 CPU с версией прошивки V4.4 при вводе параметров TO_S и CC действуют следующие правила:

- Перед каждым адресом должны быть введены пробел и открывающая угловая скобка "<".
- Закрывающая угловая скобка ">" должна быть введена после каждого адреса.
- Между адресами в TO и CC должна стоять запятая.

Ниже представлены примеры для строк параметров TO_S и CC:

- <wenna@mydomain.com>, <ruby@mydomain.com>
- <admin@mydomain.com>, <judy@mydomain.com>

Для инструкций TMAIL_C до версии 6.0 и S7-1200 CPU с версией прошивки V4.4 при вводе параметров действуют только следующие правила:

- Между адресами в TO и CC должна стоять запятая или точка с запятой.

Ниже представлены примеры для строк параметров TO_S и CC:

- wenna@mydomain.com, ruby@mydomain.com
- admin@mydomain.com, judy@mydomain.com

Для ускорения работы и экономии места в памяти инструкция TMAIL_C не выполняет проверку синтаксиса параметров TO_S и CC.

Параметры DONE, BUSY и ERROR

Выходные параметры DONE, BUSY и ERROR отображаются только в течение одного цикла, если состояние выходного параметра BUSY изменяется с "1" на "0".

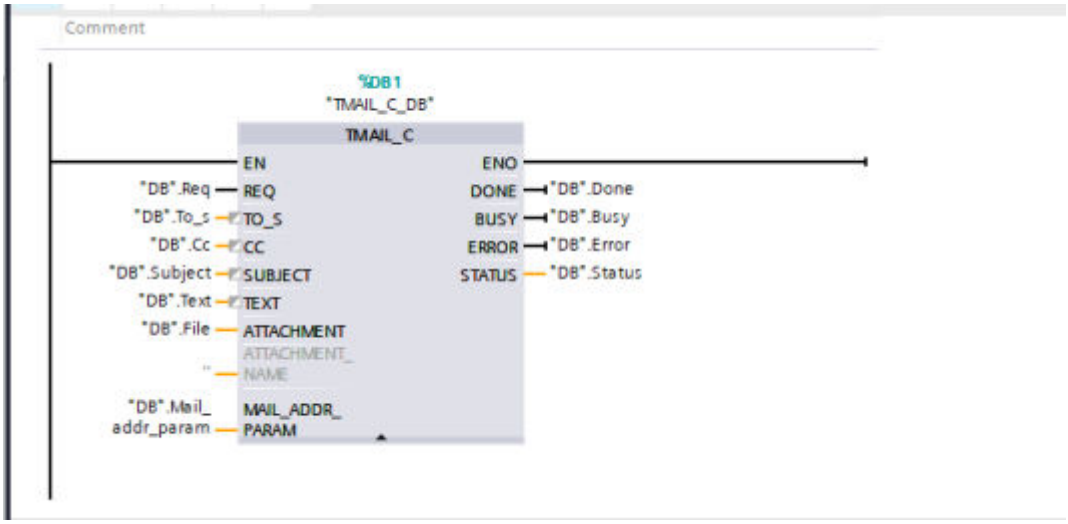
В следующей таблице представлены отношения между DONE, BUSY и ERROR. Используя эту таблицу, можно определить текущий статус инструкции TMAIL_C и момент, когда отправка электронной почты завершена.

DONE	BUSY	ERROR	Описание
0	1	0	Операция находится в обработке.
1	0	0	Операция успешно выполнена.
0	0	1	Операция завершена с ошибкой. Причина ошибки зафиксирована в параметре STATUS (Страница 908).
0	0	0	Инструкции TMAIL_C не назначено (новое) задание.

Передача архивов данных, рецептов и файлов пользователя во вложениях к электронным письмам

От TMAIL_C версии V6.0 и с прошивкой S7-1200 CPU от версии V4.4 через параметр Attachment инструкции TMAIL_C можно добавить SDT TMail_FileReference и обращаться к нему. Затем можно указать путь к файлу на карте памяти SIMATIC (SMC). Если карта памяти отсутствует, то все равно можно получить доступ к каталогам рецептов и архивов данных во внутренней загружаемой памяти PLC.

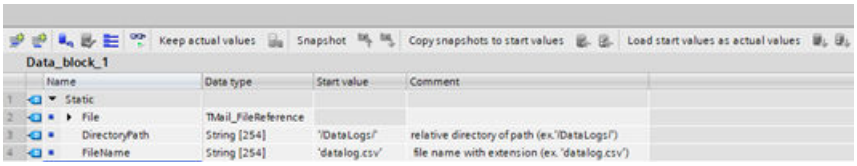
SDT TMail_FileReference автоматически использует параметр FileName для параметра AttachmentName.



SDT TMail_FileReference

SDT TMail_FileReference состоит из двух параметров и в обоих случаях речь идет о символьных строках SIMATIC:

- В параметре DirectoryPath можно адресовать каталог требуемого файла.
- Параметр FileName указывает имя файла и расширение (если применимо) файла, к которому необходимо получить доступ в каталоге, указанном предыдущим параметром.



В инструкции TMAIL_C для адресации параметра DirectoryPath можно использовать только папки DataLogs, Recipes или UserFiles. Также можно обращаться к подкаталогам внутри этих каталогов.

Помимо вышеуказанного ограничения для базового каталога, в разделе "Правила адресации файлов" ниже описаны правила, которых необходимо придерживаться при адресации подкаталогов и имен файлов.

Размер файла, который можно отправить, не ограничивается инструкцией TMAIL_C. Это следует иметь ввиду при конфигурировании своей программы.

Правила для адресации файлов

Существуют определенные правила, которым необходимо следовать, чтобы правильно адресовать файл с помощью инструкции TMAIL_C через SDT TMail_FileReference . В следующих подразделах описываются особенности для параметров DirectoryPath и FileName соответственно. В общем случае к обоим параметрам в SDT TMail_FileReference применяются следующие правила, несоблюдение которых приводит к тому, что инструкция TMAIL_C выдает статус ошибки:

- Нельзя указать пустую строку в качестве подкаталога или имени файла.
- Нельзя использовать ни в какой строке параметров управляющие символы ASCII (шестнадцатеричный диапазон: от 0x00 до 0x1F).
- Нельзя использовать ни в какой строке параметров следующие зарезервированные символы:
  - < (меньше чем)
  - > (больше чем)
  - : (двоеточие)
  - " (двойные кавычки)
  - / (косая черта) (Этот символ разрешен в DirectoryPath в качестве разделительного символа.)
  - \ (обратная косая черта)
  - | (вертикальный штрих)
  - ? (вопросительный знак)
  - * (астериск)
- Подкаталог или имя файла не могут заканчиваться пробелом или точкой.

DirectoryPath

При вводе нужного каталога в параметр DirectoryPath SDT следует учитывать следующее. Корневой каталог определяется логикой прошивки PLC и не является значимым для пользователя. Можно дополнительно ввести начальную и конечную косую черту (/). Если любая косая черта будет опущена, прошивка добавит их автоматически. Таким образом, каждая из следующих записей для DirectoryPath является правильной:

- /DataLogs/
- /DataLogs
- DataLogs/
- DataLogs

Кроме этого, можно получить доступ к уровням с глубиной вложенности большей, чем допускает корневой каталог. Для этого используется формат '/DataLogs/dir1/', где каждая косая черта (/) обозначает новый каталог. Максимальная глубина вложенности составляет восемь уровней, включая корневой каталог.

В дополнение к правилам, изложенным в разделе "Правила для адресации файлов", следует помнить, что использование относительных путей строго запрещено. Таким образом, '/DataLogs/' является недопустимым именем пути. Кроме этого, использование точки в любом компоненте подкаталога для представления текущего каталога запрещено (например, '/ DataLogs /').

FileName

При использовании параметра FileName SDT следует учитывать правила, изложенные в разделе "Правила для адресации файлов". В дополнение к этому следует помнить, что операционная система PLC ограничивает имя файла до 60 символов. При попытке обратиться к имени файла, большему или равному 60 символам, инструкция TMAIL_C прерывает свою работу и выдает ошибку.

Помимо этих исключений, можно прикрепить любой файл, независимо от его размера или расширения. Адресованный файл может включать или не включать расширение файла.

Коды ошибок

Параметр STATUS

В приведенной ниже таблице показаны возвращаемые значения TMAIL_C в параметре STATUS:

Возвращаемое значение STATUS* (W#16#...):	**Значение**	**Примечания**
0000	Выполнение TMAIL_C было завершено без ошибок	Безошибочное завершение TMAIL_C не означает, что посланное электронное письмо обязательно будет получено. Ввод неверного адреса получателя не приводит к возникновению ошибки состояния инструкции TMAIL_C. В этом случае нет никакой гарантии, что электронное письмо не будет послано другим адресатам, даже если они были введены правильно.
7001	TMAIL_C активна (BUSY = 1).	Первый вызов: Операция запущена.
7002	TMAIL_C активна (BUSY = 1).	Промежуточный вызов: Операция уже запущена.
8xxx	Обработка TMAIL_C завершена с кодом ошибки вызываемых системой коммуникационных инструкций.	Подробная информация приведена в описаниях параметра STATUS для коммуникационных инструкций TCON, TDISCON, TSEND und TRCV (Страница 864).
8009	Ошибка внутренней функции	Внутренняя функция вернула ошибку. Более подробную информацию можно найти в параметре SFB_STATUS блока данных экземпляра. Возможные значения представлены ниже:
8010	Ошибка при установлении соединения	Дополнительная информация по анализу приведена в параметре SFB_STATUS блока данных экземпляра. Указанный в параметре SFB_STATUS код ошибки объясняется в описании параметра STATUS для инструкции TCON (Страница 864).
8011	Ошибка при отправке данных	Дополнительная информация по анализу приведена в параметре SFB_STATUS блока данных экземпляра. Указанный в параметре SFB_STATUS код ошибки объясняется в описании параметра STATUS для инструкции TSEND (Страница 864).
8012	Ошибка при получении данных	Дополнительная информация по анализу приведена в параметре SFB_STATUS блока данных экземпляра. Указанный в параметре SFB_STATUS код ошибки объясняется в описании параметра STATUS для инструкции TRCV (Страница 864).

Возвращаемое значение STATUS* (W#16#...):	**Значение**	**Примечания**
8013	Ошибка при установлении соединения	Дополнительная информация по анализу приведена в параметре SFB_STATUS блока данных экземпляра. Указанный в параметре SFB_STATUS код ошибки объясняется в описании параметра STATUS для инструкций TCON (Страница 864) и TDISCON (Страница 864).
8014	Невозможно установить соединение.	Возможно был введен неверный IP-адрес почтового сервера (MailServerAddress (Страница 896)) или слишком малый промежуток времени (WatchDogTime (Страница 896)) для установления соединения. Также возможно, что CPU не подключен к сети или имеет место ошибка в конфигурации CPU.
8015	Неверный тип данных для MAIL_ADDR_PARAM	Разрешенными типами данных являются только типы системных данных (структуры) Tmail_v4 и TMail_FQDN.
8016	Неверный тип данных для параметра ATTACHMENT	Правильные типы данных перечислены в следующем списке: • ArrayOfChar • ArrayOfByte • ArrayOfWord • ArrayOfDWord • String Примечание: Типы данных ArrayOfChar и String разрешены только начиная от версии 5.0 инструкции TMAIL_C .
8017	Неверная длина данных для параметра ATTACHMENT	Длина данных должна быть <= 65534 байт.
82xx, 84xx, или 85xx	Сообщение об ошибке поступает от почтового сервера и соответствует, за исключением "8", номеру ошибки в протоколе SMTP. В следующих строках перечислены возможные коды ошибок.	Дополнительную информацию о кодах ошибок SMTP и о других кодах ошибок в протоколе SMTP можно найти в сети Интернет или в документации об ошибках почтового сервера. Кроме этого, можно просмотреть последнее отправленное почтовым сервером сообщение об ошибке в блоке данных экземпляра в параметре BUFFER1. Последние переданные с помощью инструкции TMAIL_C данные можно найти в DATEN в блоке данных экземпляра.
8450	Процесс не выполнен: Почтовый ящик не используется/недоступен	Повторить процесс позднее.
8451	Процесс отменен: Локальная ошибка обработки	Повторить процесс позднее.

Возвращаемое значение STATUS* (W#16#...):	**Значение**	**Примечания**
8500	Синтаксическая ошибка: Ошибка не распознана. Сюда же относится ошибка из-за слишком длинной командной строки. Ошибка может возникнуть и тогда, когда почтовый сервер не поддерживает процедуру аутентификации LOGIN.	Проверить параметры TMAIL_C. Попробовать отправить письмо без аутентификации. Для этого заменить содержание параметра UserName пустой строкой. Если имя пользователя не указано, метод аутентификации LOGIN также не применяется.
8501	Синтаксическая ошибка: Неверные данные в параметре	Возможная причина: Неверный адрес в параметре TO_S или CC (см. также: Параметры TO_S и CC (Страница 904)).
8502	Команда неизвестна или не реализована.	Проверить введенные данные, в частности параметр FROM. Возможно, что он неполный и не введен символ "@" или "." (см. также: Параметры TO_S и CC (Страница 904)).
8535	Неполная аутентификация SMTP	Возможно, было введено неправильное имя пользователя или неправильный пароль.
8550	Почтовый сервер недоступен. Отсутствуют права доступа.	Возможно, было введено неправильное имя пользователя или неправильный пароль или почтовый сервер не поддерживает указанной регистрации. Другой причиной может быть ошибка в доменном имени после "@" в параметре TO_S или CC (см. также: Параметры TO_S и CC (Страница 904)).
8552	Процесс отменен: Выделенное место в памяти превышено	Повторить процесс позднее.
8554	Передача не удалась	Повторить процесс позднее.
* Коды ошибок могут отображаться в редакторе текстов программ в виде целых чисел или шестнадцатеричных значений.		

Параметр SFB_STATUS блока данных экземпляра

От версии 4.4 прошивки S7-1200 с версией 6.0 или выше инструкции TMAIL_C возможны следующие возвращаемые значения в параметре SFB_STATUS DB экземпляра:

Возвращаемые значения в параметре SFB_STATUS DB экземпляра (W#16#...)	**Значение**
8085	Идентификатор соединения (параметр ID) уже использует сконфигурированным соединением.
8086	Параметр ID находится вне допустимого диапазона.
8087	Достигнуто максимальное число соединений; дальнейшие соединения невозможны.
8088 *	Файл отсутствует или недоступен в настоящий момент.

Возвращаемые значения в параметре SFB_STATUS DB экземпляра (W#16#...)	**Значение**
8089 *	Файл не может быть открыт, так как число одновременно открытых файлов превысило системное предельное значение. Для S7-1200 предельное значение составляет 26 файлов в файловой системе.
808A *	DirectoryPath содержит каталог, отличный от DataLogs, Recipes или UserFiles, или один из адресованных подкаталогов нарушает ранее упомянутые правила адресации. Дополнительную информацию см. в DirectoryPath (Страница 905).
808B *	FileName содержит недопустимую строку символов или оставлен пустым. Дополнительную информацию см. в FileName (Страница 905).
808C *	Параметр AttachmentName при адресации пути к файлу как вложению должен быть пустым.
8092	Параметры TO_S и CC пустые или дополнительный параметр "From" пустой или не полный.
8093	Параметр MAIL_ADDR_PARAM требуется обновления до защищенного соединения, но почтовый сервер не поддерживает команду STARTTLS.
8095	Недействительный ответ от почтового сервера. Возможно, что почтовый сервер не поддерживает RFC.
809A	Структура SDT в параметре MAIL_ADDR_PARAM не поддерживается на встроенном интерфейсе.
809B	Недействительный интерфейсный ID в SDT в параметре MAIL_ADDR_PARAM.
80A1	Указанное соединение или удаленный порт уже используются.
80A3	Созданное пользовательской программой соединение использует ID.
80A4	Недопустимый IP-адрес удаленной конечной точки соединения, или он совпадает с IP-адресом локального участника.
80A7	Ошибка обмена данными: Инструкция TDISCON выполнена до завершения TMAIL_C.
80B7	Удаленный порт = 0 или для IP-адреса конечной точки участника было установлено 0.0.0.0.
80C3	Недостаточно ресурсов в CPU
80C4	Временная ошибка обмена данными: • В настоящий момент соединение не может быть установлено. • Соединение не может быть установлено, так как брандмауэры по пути соединения не открыты для требуемых портов. • Интерфейс находится в процессе приема новых параметров. • В данный момент выполняется удаление сконфигурированного соединения инструкцией TDISCON.

Возвращаемые значения в параметре SFB_STATUS DB экземпляра (W#16#...)	**Значение**
80C5	Почтовый сервер отклоняет установление соединения, завершил соединение или выполнил его активную отмену.
80C6	Участник соединения не готов к приему (ошибка сети).
80C7	Превышение времени при выполнении
80C8	Попытка заново установить имеющееся соединение.
80C9	Валидация участника соединения не удалась. Почтовый сервер не соответствует установленному в параметре MailServerAddress участнику.
80CE	IP-адрес локального интерфейса 0.0.0.0.
80D0	Параметр MailServerAddress содержит при использовании DNS пустую строку символов.
80D1	Параметр MailServerAddress не является полным доменным именем. Возможно, не хватает точки в конце.
80D2	Адрес DNS-сервера не был настроен.
80D3	Не удалось разрешить полное доменное имя (FQDN). Возможные причины: • DNS-сервер недоступен (пример: DNS-сервер отключен или удаленный порт недоступен). • При связи с DNS-сервером возникла ошибка. • DNS-сервер вернул действительный DNS-ответ, но в нем нет IPv4-адреса.
80E0	Связь с почтовым сервером прервалась из-за сообщения с ошибками. Возможные причины: • Недействительный код аутентификации сообщения • Не удалось расшифровать сообщение • Ошибка при распаковке сообщения • Переполнение внутренней емкости
80E1	Ошибка при подтверждении. Возможные причины: • Отмена со стороны пользователя • Недостаточный уровень безопасности. • Отсроченное согласование не поддерживается • SSL/TLS версия не поддерживается • Ошибка декодирования
80E2	Сертификат не поддерживается / недействителен Возможная причина: Время у затронутого модуля не установлено или модуль не синхронизирован. Пример: Стандартная настройка для даты модуля 1/1/2012, и дата не была установлена при вводе в эксплуатацию. Срок действия сертификата начинается 20 августа 2016 г. и завершается 20 августа 2024 г. В этом случае дата модуля не соответствует сроку действия сертификата и поэтому сертификат считается недействительным для модуля.
80E3	Сертификат почтового сервера был отклонен.

Возвращаемые значения в параметре SFB_STATUS DB экземпляра (W#16#...)	Значение
80E4	Не найдет действительный орган сертификации для сертификата почтового сервера.
80E5	Сертификат почтового сервера истек.
80E6	Ошибки целостности в протоколе Transport-Layer-Security
80E7	Не поддерживаемое расширение в сертификате почтового сервера
80E9	TLS-сервер без сертификата сервера не поддерживается

* Эти коды ошибок добавляются для упрощения диагностики при неправильной адресации пути к файлу инструкции TMAIL_C.

11.2.8.15 UDP

UDP (User Datagram Protocol) это описанный в RFC 768 стандартный протокол передачи дейтаграмм пользователя. UDP предлагает механизм для передачи дейтаграмм между приложениями. Но доставка данных не гарантируется. Этот протокол обладает следующими особенностями:

- Быстрый протокол коммуникации
- Подходит для данных небольшого и среднего объема (до 1472 байтов)
- UDP - это более простой, чем TCP, тонкослойный транспортный протокол с более низкими накладными расходами
- Может использоваться очень гибко со сторонними системами
- Поддерживает маршрутизацию
- Направляет дейтаграммы через номера портов
- Сообщения не квитируются: Приложение отвечает за восстановление после ошибки и за безопасность
- Для управления данными используется программирование из-за интерфейса программирования SEND / RECEIVE

UDP поддерживает широковещательную передачу. Чтобы использовать широковещательную передачу, необходимо сконфигурировать часть адреса IP в конфигурации ADDR. Пример: CPU с IP-адресом 192.168.2.10 и маской подсети 255.255.255.0 использовал бы широковещательный адрес 192.168.2.255.

11.2.8.16 TUSEND и TURCV

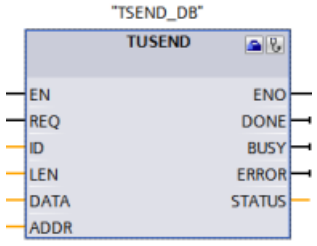
Следующие инструкции управляют процессом UDP коммуникации:

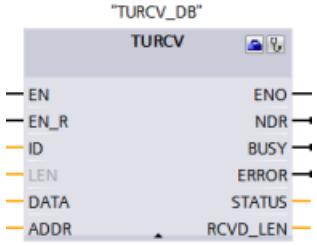
- Инструкция TCON устанавливает связь между клиентским и серверным компьютером (CPU).
- Инструкции TUSEND и TURCV передают и принимают данные.
- Инструкция TDISCON разрывает связь между клиентом и сервером.

Дополнительную информацию о коммуникационных инструкциях TCON и TDISCON можно найти в разделе "TCP и ISO-on-TCP" в TCON, TDISCON, TSEND и TRCV (Страница 864).

Таблица 11-45 Инструкции TUSEND и TURCV

LAD/FBD	SCL	Описание

	`"TUSEND_DB"(`
	`req:=_bool_in_,`
	`ID:=_word_in_,`
	`len:=_udint_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`data:=_variant_inout_);`
		Инструкция TUSEND отправляет данные по протоколу UDP указанному с помощью параметра ADDR удаленному участнику.
		Для запуска задания на передачу данных, вызвать инструкцию TUSEND с REQ = 1.

	`"TURCV_DB"(`
	`en_r:=_bool_in_,`
	`ID:=_word_in_,`
	`len:=_udint_in_,`
	`ndr=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`rcvd_len=>_udint_out_,`
	`data:=_variant_inout_);`
		Инструкция TURCV получает данные по UDP. Параметр ADDR показывает адрес отправителя. После успешного завершения TURCV параметр ADDR содержит адрес удаленного участника (отправителя).
		TURCV не поддерживает Ad-hoc режим.
		Для запуска задания на прием данных, вызвать инструкцию TURCV с EN_R = 1.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Инструкции TCON, TDISCON, TUSEND и TURCV выполняются асинхронно, поэтому обработка одной операции происходит на несколько выполнений инструкции.

Таблица 11-46 Типы данных для параметров TUSEND и TURCV

Параметр и тип данных		Тип данных	Описание
REQ (TUSEND)	IN	Bool	Запускает операцию передачи по положительному фронту. Данные передаются из указанной с помощью DATA и LEN области.
EN_R (TURCV)	IN	Bool	• 0: CPU не может выполнить прием. • 1: Активирует CPU для приема. Инструкция TURCV готова к приему и задание приема обрабатывается.
ID	IN	Word	Ссылка на соответствующее соединение между прикладной программой и сеансовым уровнем операционной системы. ID должен совпадать с соответствующим ID параметра в локальном описании соединения. Диапазон значений: От W#16#0001 до W#16#0FFF.
LEN	IN	UDInt	Количество байтов, которые должны быть отправлены (TUSEND) или получены (TURCV). • По умолчанию = 0. Параметр DATA определяет длину передаваемых или принимаемых данных. • В остальном, диапазон значений: От 1 до 1472

Параметр и тип данных		Тип данных	Описание
DONE (TUSEND)	IN	Bool	Параметр состояния DONE (TUSEND): • 0: Операция еще не запущена или еще выполняется. • 1: Операция выполнена без ошибок.
NDR (TURCV)	OUT	Bool	Параметр состояния NDR (TURCV): • 0: Операция еще не запущена или еще выполняется. • 1: Задание успешно выполнено.
BUSY	OUT	Bool	• 1: Операция еще не завершена. Новая операция не может быть запушена. • 0: Операция завершена.
ERROR	OUT	Bool	Параметры состояния со следующими значениями: • 0: Ошибки отсутствуют • 1: Ошибка при обработке. STATUS содержит подробную информацию о типе ошибки.
STATUS	OUT	Word	Информация о состоянии включает в себя информацию об ошибке. (См. таблицу "Коды условий для ERROR и STATUS" ниже.)
RCVD_LEN	OUT	UDInt	Количество принятых байт (TURCV)
DATA	IN_OUT	Variant	Адрес области передачи (TUSEND) или области приема (TURCV): • образ процесса входов • образ процесса выходов • меркер-бит • блок данных
ADDR	IN_OUT	Variant	Указатель на адрес получателя (для TUSEND) или отправителя (для TURCV) (например, P#DB100.DBX0.0 байт 8). Указатель может ссылаться на любую область памяти. Требуется следующая структура из 8 байтов: • Первые 4 байта содержат удаленный IP-адрес. • Следующие 2 байта определяют номер удаленного порта. • Последние 2 байта зарезервированы.

Состояние задания отображается в выходных параметрах BUSY и STATUS. STATUS соответствует выходному параметру RET_VAL у асинхронно работающих инструкций.

В следующей таблице представлены отношения между BUSY, DONE (TUSEND), NDR (TURCV) и ERROR. Используя эту таблицу, можно определить текущий статус инструкции (TUSEND или TURCV) и момент, когда отправка или получение электронной почты завершены.

Таблица 11-47 Состояние параметров BUSY, DONE (TUSEND)/NDR (TURCV) и ERROR

BUSY	DONE / NDR	ERROR	Описание
TRUE	Не имеет значения	Не имеет значения	Операция находится в обработке.
FALSE	TRUE	FALSE	Операция была успешно выполнена.

BUSY	DONE / NDR	ERROR	Описание
FALSE	FALSE	TRUE	Операция завершена с ошибкой. Причина ошибки зафиксирована в параметре STATUS.
FALSE	FALSE	FALSE	Инструкции не назначено (новое) задание.

[1] Следствие асинхронной работы инструкций: Для TUSEND данные в области передачи должны оставались непротиворечивыми, пока один из параметров DONE или ERROR не примет значение TRUE. Для TURCV данные в области приема непротиворечивы только в том случае, если параметр NDR = TRUE.

Таблица 11-48 Коды условий TUSEND иTURCV для ERROR и STATUS

ERROR	STATUS	Описание
0	0000	• Операция передачи была выполнена без ошибок (TUSEND). • Новые данные были приняты. Фактическая длина полученных данных отображается в RCVD_LEN (TURCV).
0	7000	• Нет активных операций (TUSEND) • Блок не готов к получению (TURCV)
0	7001	• Запустить обработку операции, идет отправка данных (TUSEND): Во время обработки операционная система обращается к данным в области отправки DATA. • Блок готов к приему, операция приема была активирована (TURCV).
0	7002	• Следующий вызов (REQ не имеет значения), операция находится в обработке (TUSEND):: Во время обработки операционная система обращается к данным в области отправки DATA. • Следующий вызов, операция находится в обработке: Во время обработки инструкция TURCV записывает данные в область приема. Поэтому возникновение ошибки может привести появлению противоречивых данных в области приема.
1	8085	Параметр LEN больше максимально допустимого значения, имеет значение 0 (TUSEND) или значение параметра LEN или DATA было изменено после последнего выполнения инструкции (TURCV).
1	8086	Параметр ID за пределами допустимой области адресов.
1	8088	• Параметр LEN больше, чем указанная в DATA область памяти (TUSEND) или область приема (TURCV). • Слишком маленькая область получения (TURCV).
1	8089	Параметр ADDR ссылается не на блок данных.
1	80A1	Ошибка обмена данными: • Указанное соединение между программой пользователя и сеансовым уровнем операционной системы еще не было установлено. • Указанное соединение между программой пользователя и сеансовым уровнем операционной системы завершается в настоящий момент. Операция передачи (TUSEND) или приема (TURCV) или приема по этому соединению невозможна. • Интерфейс инициализируется заново.
1	80A4	Недействительный IP-адрес удаленной точки соединения, возможно он соответствует локальному IP-адресу (TUSEND).

ERROR	STATUS	Описание
1	80B3	• Установленный вариант протокола (параметр connection_type в описании соединения) не является протоколом UDP. Просьба использовать инструкцию TSEND или TRCV. • Параметр ADDR: Неправильный настройки для номера порта (TUSEND)
1	80C3	• Блок с таким ID уже обрабатывается в другом классе приоритетов. • Внутренний дефицит ресурсов
1	80C4	Временная ошибка обмена данными: • Соединение между программой пользователя и сеансовым уровнем операционной системы не может быть установлено сейчас (TUSEND). • Интерфейс находится в процессе приема новых параметров (TUSEND). • Указанное соединение устанавливается (TURCV).

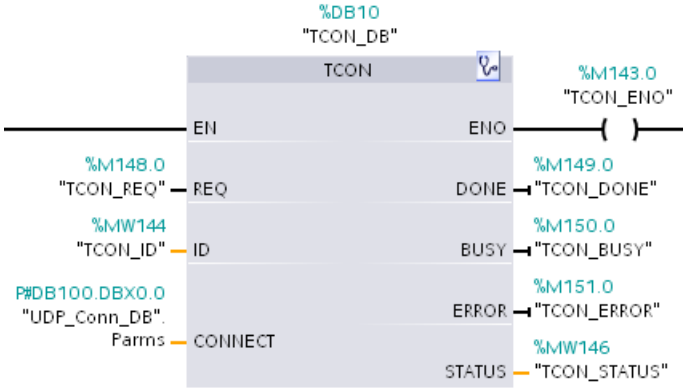
Протоколы соединения Ethernet

У каждого CPU есть встроенный порт PROFINET с поддержкой стандартной PROFINET коммуникации. Инструкции TUSEND и TURCV поддерживают UDP Ethernet протокол.

Дополнительную информацию см. в главе "Конфигурация устройства" в "Конфигурирование пути соединения между локальным и партнерским CPU" (Страница 812).

Режим работы

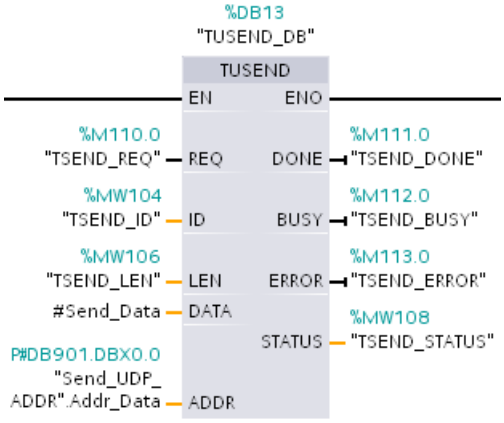
Оба участника в UDP коммуникациях являются пассивными. Типовые стартовые значения для параметров с типом данных "TCON_Param" представлены на рисунках ниже. Номера портов (LOCAL_TSAP_ID) записаны в 2-байтовом формате. Все порты за исключением 161, 34962, 34963, и 34964 разрешены.



UDP_Conn_DB

	Name	Data type	Offset	Initial value	Comment
1	Static				
2	Parms	TCON_Param	0.0		
3	BLOCK_LENGTH	UInt	0.0	64	byte length of SDT
4	ID	CONN_OUC	2.0	1	reference to the connection
5	CONNECTION_TYPE	USInt	4.0	19	17: TCP/IP, 18: ISO on TCP
6	ACTIVE_EST	Bool	5.0	false	active/passive connection establishment
7	LOCAL_DEVICE_ID	USInt	6.0	1	1: local IE interface
8	LOCAL_TSAP_ID_L...	USInt	7.0	2	byte length of local TSAP id/port number
9	REM_SUBNET_ID_...	USInt	8.0	0	byte length of remote subnet id
10	REM_STADDR_LEN	USInt	9.0	0	byte length of remote IP address
11	REM_TSAP_ID_LEN	USInt	10.0	0	byte length of remote port/TSAP id
12	NEXT_STADDR_LEN	USInt	11.0	0	byte length of next station address
13	LOCAL_TSAP_ID	Array[1..16] of Byte	12.0		TSAP id/local port number
14	LOCAL_TSAP_I...	Byte		B#16#07	
15	LOCAL_TSAP_I...	Byte		B#16#D0	

Инструкция TUSEND отправляет данные по протоколу UDP указанному с помощью параметра TADDR_Param удаленному участнику. Инструкция TURCV получает данные по UDP. После успешного выполнения инструкции TURCV тип данных "TADDR_Param" содержит адрес удаленного участника (отправителя).



Send_UDP_ADDR

	Name	Data type	Offset	Initial value	Comment
1	Static				
2	Addr_Data	TADDR_Param	0.0		
3	REM_IP_ADDR	Array[1..4] of USint	0.0		remote station address
4	REM_IP_ADDR[1]	USInt		192	
5	REM_IP_ADDR[2]	USInt		168	
6	REM_IP_ADDR[3]	USInt		2	
7	REM_IP_ADDR[4]	USInt		10	
8	REM_PORT_NR	UInt	4.0	2000	remote port number
9	RESERVED	Word	6.0	0	ununsed; has to be 0

11.2.8.17 T_CONFIG

Инструкция T_CONFIG может через программу пользователя изменить Ethernet адрес, имя устройства PROFINET или IP-адрес NTP-сервера для синхронизации времени. Возможно постоянное или временное изменение следующих функций:

- IP-адрес
- Маска подсети
- Адрес маршрутизатора
- Имя станции
- IP-адреса до четырех серверов NTP

Примечание

На странице "Ethernet адрес" в "Свойствах" CPU с помощью радиокнопки "IP-адрес устанавливается непосредственно на устройстве" (Страница 926) после загрузки программы в CPU можно изменить IP-адрес онлайн или через инструкцию "T_CONFIG".

На странице "Ethernet адрес" в "Свойствах" CPU с помощью радиокнопки "Имя устройства PROFINET устанавливается непосредственно на устройстве" (Страница 928) после загрузки программы в CPU можно изменить имя устройства PROFINET онлайн или через инструкцию "T_CONFIG".

На странице "Синхронизация времени" в "Свойствах" CPU через поле "Активировать синхронизацию времени через NTP-сервер" (Страница 929) возможно изменение IP-адресов до четырех серверов NTP.

Примечание

Одновременное выполнение нескольких инструкций T_CONFIG невозможно.

Примечание

Изменения в IP-адресе или в имени станции CPU могут быть временными или постоянными. Изменения в IP-адресах NTP-сервера могут быть только временными:

- Постоянное изменение означает сохранение, т.е. изменение останется и после отключения питания.
- Временное изменение является энергозависимым, т.е. после отключения питания восстанавливается первоначальное значение.

Таблица 11-49 Инструкция T_CONFIG

LAD/FBD	SCL	Описание

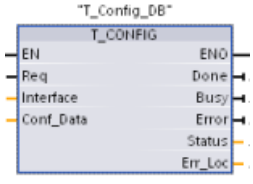
	`"T_CONFIG_DB"(`
	`  Req:=_bool_in_,`
	`  Interface:=_uint_in_,`
	`  Conf_Data:=_variant_in_,`
	`  Done=>_bool_out_,`
	`  Busy=>_bool_out_,`
	`  Error=>_bool_out_,`
	`  Status=>_dword_out_,`
	`  Err_Loc=>_dword_out_);`
		Инструкция T_CONFIG изменяет параметры IP-конфигурации через программу пользователя.
		Инструкция T_CONFIG работает асинхронно. Обработка растягивается на несколько вызовов.

Таблица 11-50 Типы данных T_CONFIG для параметров

Параметр и тип данных		Тип данных	Описание
REQ	Input	Bool	Запускает инструкцию при положительном фронте.
INTERFACE	Input	HW_Interface	ID сетевого интерфейса
CONF_DATA	Input	Variant	Ссылка на структуры данных конфигурации; CONF_DATA определена через структуру, содержащую до четырех типов системных данных (SDT).
DONE	Output	Bool	• 0: Операция еще не запущена или еще выполняется. • 1: Операция была выполнена без ошибок.
BUSY	Output	Bool	• 0: Операция завершена. • 1: Операция еще не завершена. Новая операция не может быть запушена.
ERROR	Output	Bool	Параметры состояния со следующими значениями: • 0: Ошибки отсутствуют • 1: При обработке возникла ошибка. STATUS содержит информацию о типе ошибки.
STATUS	Output	DWord	Информация о состоянии включает в себя информацию об ошибке. (См. таблицу "Коды условий для ERROR и STATUS" ниже.)
ERR_LOC	Output	DWord	Место возникновения ошибки (ID поля и адрес подполя в структуре CONF_DATA)

Данные IP-конфигурации сохраняются в блок данных CONF_DATA, вместе с указателем Variant на указанный выше параметр CONF_DATA. Выполнение инструкции T_CONFIG успешно завершается после передачи данных IP-конфигурации на сетевой интерфейс.

Сообщения о состоянии и ошибках инструкцииT_CONFIG"выводятся в параметрах STATUS и ERR_LOC:

- Причина ошибки выводится в параметре STATUS.
- Место возникновения ошибки выводится в параметре ERR_LOC. Доступны следующие опции:
  - 16#0000_0000: Ошибки отсутствуют или ошибка при вызове инструкции (например, ошибка при настройке инструкции или в процессе обмена данными с интерфейсом PROFINET).
  - 16#0001_0000: Ошибка в данных конфигурации в параметрах типа системных данных IF_CONF_HEADER.
  - 16#0001_000x: Ошибка в данных конфигурации в параметрах типа системных данных IF_CONF_V4 или IF_CONF_NOS или IF_CONF_NTP (x указывает позицию субблока с ошибкой в структуре T_CONFIG. Например, если структура T_CONFIG содержит субблок для IP-адреса и субблок для имени станции, и ошибка находится в субблоке для имени станции, то ERR_LOC имеет значение 0001_0002.)

В таблице ниже показаны возможные значения для параметров STATUS и ERR_LOC:

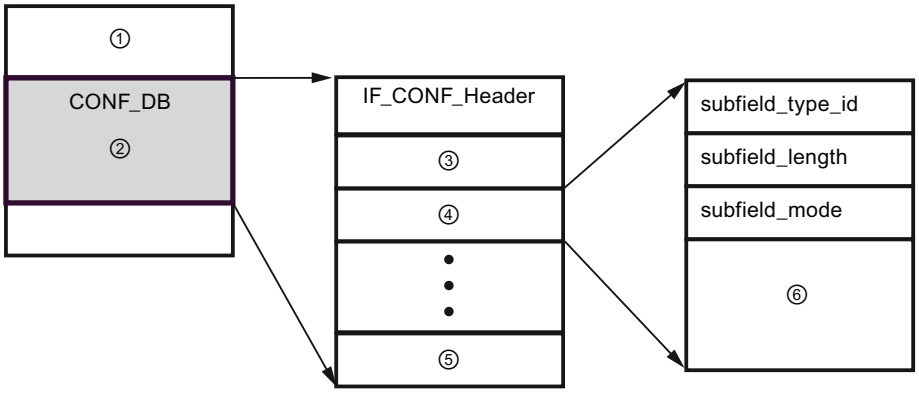
STATUS*	ERR_LOC*	Значение
0000_0000	0000_0000	Обработка задания выполнена без ошибок.
0070_0000	0000_0000	Нет активных обрабатываемых заданий.

STATUS*	ERR_LOC*	Значение
0070_0100	0000_0000	Начало обработки задания.
0070_0200	0000_0000	Промежуточный вызов (REQ не играет роли):
C08x_yy00	0000_0000	Общая информация об ошибках.
C080_8000	0000_0000	Ошибка при вызове инструкции: Недействительный аппаратный идентификатор в параметре Interface.
C080_8100	0000_0000	Ошибка при вызове инструкции: Аппаратный идентификатор в параметре Interface не адресует интерфейс PROFINET.
C080_8700	0000_0000	Ошибка при вызове инструкции: Неверная длина блока данных в параметре CONF_DATA.
C080_8800	0001_0000	Ошибка в типе системных данных IF_CONF_HEADER: Неверное значение параметра FieldType. Использовать для FieldType значение "0".
C080_8900	0001_0000	Ошибка в типе системных данных IF_CONF_HEADER: Параметр FieldId имеет неверное значение или использовался несколько раз. Использовать для FieldId значение "0".
C080_8A00	0001_0000	Ошибка в типе системных данных IF_CONF_HEADER: Неверное число в параметре SubfieldCount. Ввести правильный номер используемых типов системных данных IF_CONF_V4, IF_CONF_NOS или IF_CONF_NTP.
C080_8B00	0001_000x	Ошибка в типе системных данных IF_CONF_V4, IF_CONF_NOS или IF_CONF_NTP: Неверное значение параметра Id. Использовать для IF_CONF_V4 значение 30, для IF_CONF_NOS значение 40, дляIF_CONF_NTP значение 17.
C080_8C00	0001_000x	Ошибка в типе системных данных IF_CONF_V4, IF_CONF_NOS или IF_CONF_NTP: Используется неправильный тип системных данных, неправильная последовательность или многократное использование одного типа системных данных.
C080_8D00	0001_000x	Ошибка в типе системных данных IF_CONF_V4, IF_CONF_NOS или IF_CONF_NTP: Неверное или недействительное значение параметра Length.
C080_8E00	0001_000x	Ошибка в типе системных данных IF_CONF_V4, IF_CONF_NOS или IF_CONF_NTP: Неверное или недействительное значение параметра Mode. • Для IF_CONF_V4 и IF_CONF_NOS разрешены только значения 1 (постоянно) или 2 (временно). • Для IF_CONF_NTP допускается только значение "2" (временно).
C080_9000	0001_000x	Ошибка в типе системных данных IF_CONF_V4, IF_CONF_NOS или IF_CONF_NTP: Данные конфигурации не могут быть применены. Возможная причина: • Для IF_CONF_V4: В конфигурации оборудования не выбрана настройка "Установить IP-адрес на устройстве". • Для IF_CONF_NOS: В конфигурации оборудования не выбрана настройка "Установить имя устройства PROFINET на устройстве". • Для IF_CONF_NTP: В конфигурации оборудования не выбрана настройка "Активировать синхронизацию времени через NTP-сервер" и для NTP-сервера не был установлен IP-адрес.
C080_9400	0001_000x	Ошибка в типе системных данных IF_CONF_V4, IF_CONF_NOS или IF_CONF_NTP: Значение параметра не определено или является недействительным.
C080_9500	0001_000x	Ошибка в типе системных данных IF_CONF_V4, IF_CONF_NOS или IF_CONF_NTP: Значения двух параметров являются противоречивыми.

STATUS*	ERR_LOC*	Значение
C080_C200	0000_0000	Ошибка при вызове инструкции: Данные конфигурации не могут быть переданы. Возможная причина: Интерфейс PROFINET недоступен.
C080_C300	0000_0000	Ошибка при вызове инструкции: Недостаточно ресурсов (например, многократный вызов "T_CONFIG" с различными параметрами).
C080_C400	0000_0000	Ошибка при вызове инструкции: Временная ошибка обмена данными. Отображение часов для перехода на летнее время.
C080_D200	0000_0000	Ошибка при вызове инструкции: Вызов невозможен. Инструкция не поддерживается интерфейсом PROFINET.

Блок данных CONF_DATA

Следующая схема показывает, как данные конфигурации, которые будут переданы, сохраняются в DB конфигурации.



① DB конфигурации
② Данные конфигурации
③ Субполе 1
④ Субполе 2
⑤ Субполе *n*
⑥ Специфичные параметры субполя

Данные конфигурации CONF_DB состоят из поля с заголовком (IF_CONF_Header) и несколько субполей. IF_CONF_Header предоставляет следующие элементы:

- field_type_id (тип данных UInt): Ноль
- field_id (тип данных UInt): Ноль
- subfield_cnt (тип данных UInt): Количество субполей:

Каждое субполе состоит из заголовка (subfield_type_id, subfield_length, subfield_mode) и специфических параметров субполя. Каждое субполе должно состоять из четного числа байтов. subfield_mode может поддерживать значения 1 и 2 (см. таблицу ниже).

Примечание

В настоящее время допускается одно поле (IF_CONF_Header). Его параметры field_type_id и field_id должны быть равны 0. Дополнительные поля с другими значениями для field_type_id и field_id оставлены для будущих расширений.

Таблица 11-51 Поддерживаемые субполя

subfield_type_id	Тип данных	Значение
30	IF_CONF_V4	Параметры IP: IP-адрес, маска подсети, адрес маршрутизатора
40	IF_CONF_NOS	Имя устройства PROFINET IO (Name of station)
17	IF_CONF_NTP	Network Time Protocol (NTP)

Таблица 11-52 Элементы типа данных IF_CONF_V4

Имя		Тип данных	Стартовое значение	Описание
Id		UInt	30	subfield_type_id
Length		UInt	18	subfield_length
Mode		UInt	0	subfield_mode (1: постоянно или 2: временно)
InterfaceAddress		IP_V4	-	Адрес интерфейса
ADDR		Array [1..4] of Byte		
	ADDR[1]	Byte	0	Старший байт IP-адреса: 200
	ADDR[2]	Byte	0	Старший байт IP-адреса: 12
	ADDR[3]	Byte	0	Младший байт IP-адреса: 1
	ADDR[4]	Byte	0	Младший байт IP-адреса: 144
SubnetMask		IP_V4	-	Маска подсети
ADDR		Array [1..4] of Byte		
	ADDR[1]	Byte	0	Старший байт маски подсети: 255
	ADDR[2]	Byte	0	Старший байт маски подсети: 255
	ADDR[3]	Byte	0	Старший байт маски подсети: 255
	ADDR[4]	Byte	0	Старший байт маски подсети: 0
DefaultRouter		IP_V4	-	Маршрутизатор по умолчанию
ADDR		Array [1..4] of Byte		
	ADDR[1]	Byte	0	Старший байт маршрутизатора: 200
	ADDR[2]	Byte	0	Старший байт маршрутизатора: 12
	ADDR[3]	Byte	0	Младший байт маршрутизатора: 1
	ADDR[4]	Byte	0	Младший байт маршрутизатора: 1

Таблица 11-53 Элементы типа данных IF_CONF_NOS

Имя	Тип данных	Стартовое значение	Описание
Id	UInt	40	subfield_type_id
Length	UInt	246	subfield_length

Имя	Тип данных	Стартовое значение	Описание
Mode	UInt	0	subfield_mode (1: постоянно или 2: временно)
NOS (Name of station)	Array[1..240] of Byte	0	Имя станции: Необходимо определить ARRAY с первого байта. Если ARRAY имеет большую длину, чем у присваиваемого имени станции, то необходимо после имени станции ввести нулевой байт (соответствует IEC 61158-6-10). В противном случае NOS отклоняется и инструкция "T_CONFIG (Страница 919)" вносит код ошибки DW#16#C0809400 в STATUS. Если первый байт занимается 0, то имя станции удаляется.

Дл имени станции действуют следующие ограничения:

- Компонент имени в имени станции, т.е., строка символов между двумя точками, не должен быть больше 63 символов.
- Никакие специальные символы, такие как умляуты, скобки, подчеркивание, наклонная черта, пробел, и т.д. не разрешаются. Единственным разрешенным специальным символом является тире.
- Имя станции не должно начинаться с символа "-" или заканчиваться им.
- Имя станции не должно начинаться с цифры.
- Формат имени станции n.n.n.n (n = 0... 999) не разрешен.
- Имя станции не должно начинаться последовательностью символов "port-xyz" или "port-xyz-abcde" (a, b, c, d, e, x, y, z = 0, . 9).

Примечание

Также можно создать ARRAY "NOS" с длиной меньше 240 байт. Но минимальная длина составляет 2 байта. В этом случае необходимо соответствующим образом настроить переменную "Length" (длина субполя).

Таблица 11-54 Элементы типа данных IF_CONF_NTP

Имя	Тип данных	Стартовое значение	Описание
Id	UInt	17	subfield_type_id
Length	UInt	22	subfield_length
Mode	UInt	0	subfield_mode (2: временно)
NTP_IP	Array[1...4] of IP_V4	-	IP-адреса NTP-серверов
NTP_IP[1]	IP_V4		IP-адреса NTP-сервера 1
ADDR	Array[1...4] of Byte	0	
ADDR[1]	Byte	0	Старший байт IP-адреса
ADDR[2]	Byte	0	Старший байт IP-адреса
ADDR[3]	Byte	0	Младший байт IP-адреса
ADDR[4]	Byte	0	Младший байт IP-адреса
NTP_IP[2]	IP_V4		IP-адреса NTP-сервера 2
ADDR	Array[1...4] of Byte	0	

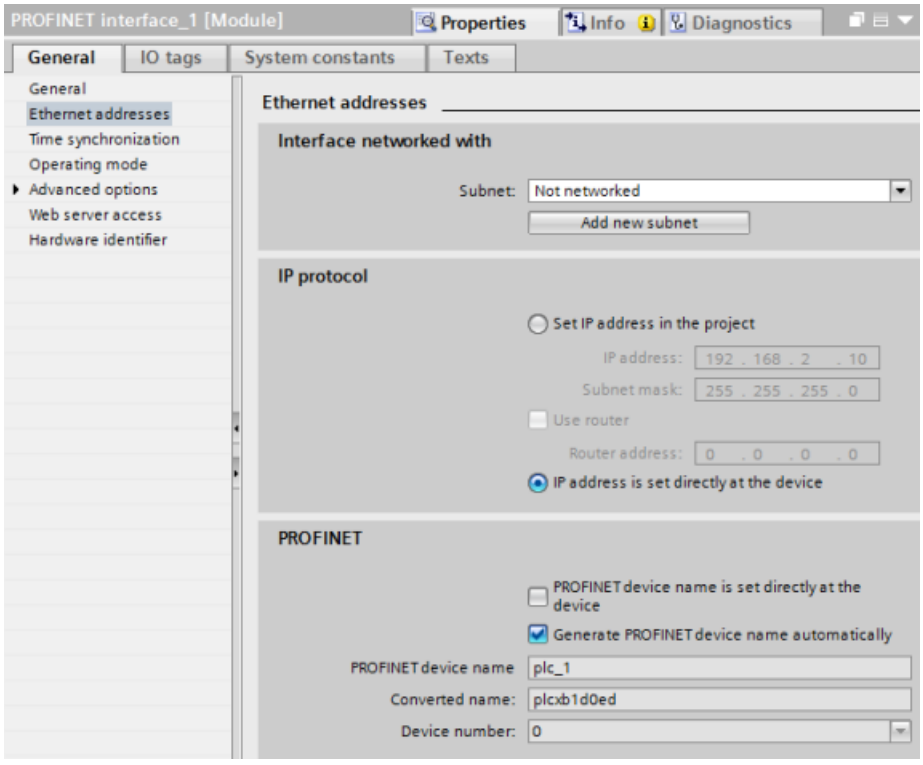
Имя				Тип данных	Стартовое значение	Описание
			ADDR[1]	Byte	0	Старший байт IP-адреса
			ADDR[2]	Byte	0	Старший байт IP-адреса
			ADDR[3]	Byte	0	Младший байт IP-адреса
			ADDR[4]	Byte	0	Младший байт IP-адреса
	NTP_IP[3]			IP_V4		IP-адреса NTP-сервера 3
		ADDR		Array[1...4] of Byte	0	
			ADDR[1]	Byte	0	Старший байт IP-адреса
			ADDR[2]	Byte	0	Старший байт IP-адреса
			ADDR[3]	Byte	0	Младший байт IP-адреса
			ADDR[4]	Byte	0	Младший байт IP-адреса
	NTP_IP[4]			IP_V4		IP-адреса NTP-сервера 4
		ADDR		Array[1...4] of Byte	0	
			ADDR[1]	Byte	0	Старший байт IP-адреса
			ADDR[2]	Byte	0	Старший байт IP-адреса
			ADDR[3]	Byte	0	Младший байт IP-адреса
			ADDR[4]	Byte	0	Младший байт IP-адреса

Пример: Изменение параметров IP с помощью инструкции T_CONFIG

В следующем примере в субполе "addr" будут изменены параметры "InterfaceAddress" (IP-адрес), "SubnetMask" и "DefaultRouter" (IP-маршрутизатор). На странице "Ethernet адрес" в "Свойствах" CPU с помощью радиокнопки "IP-адрес устанавливается непосредственно на устройстве" после загрузки программы в CPU можно изменить IP-параметр через инструкцию "T_CONFIG".

CONF_DATA_1

	Name	Data type	Start value
1	Static		
2	Conf_data	Struct	
3	header	IF_CONF_Header	
4	FieldType	UInt	0
5	FieldId	UInt	0
6	SubfieldCount	UInt	1
7	addr	IF_CONF_v4	
8	Id	UInt	30
9	Length	UInt	18
10	Mode	UInt	1
11	InterfaceAddress	IP_V4	
12	ADDR	array [1..4] of Byte	
13	ADDR[1]	Byte	192
14	ADDR[2]	Byte	168
15	ADDR[3]	Byte	2
16	ADDR[4]	Byte	30
17	SubnetMask	IP_V4	
18	ADDR	array [1..4] of Byte	
19	ADDR[1]	Byte	255
20	ADDR[2]	Byte	255
21	ADDR[3]	Byte	255
22	ADDR[4]	Byte	0
23	DefaultRouter	IP_V4	
24	ADDR	array [1..4] of Byte	
25	ADDR[1]	Byte	192
26	ADDR[2]	Byte	168
27	ADDR[3]	Byte	2
28	ADDR[4]	Byte	1

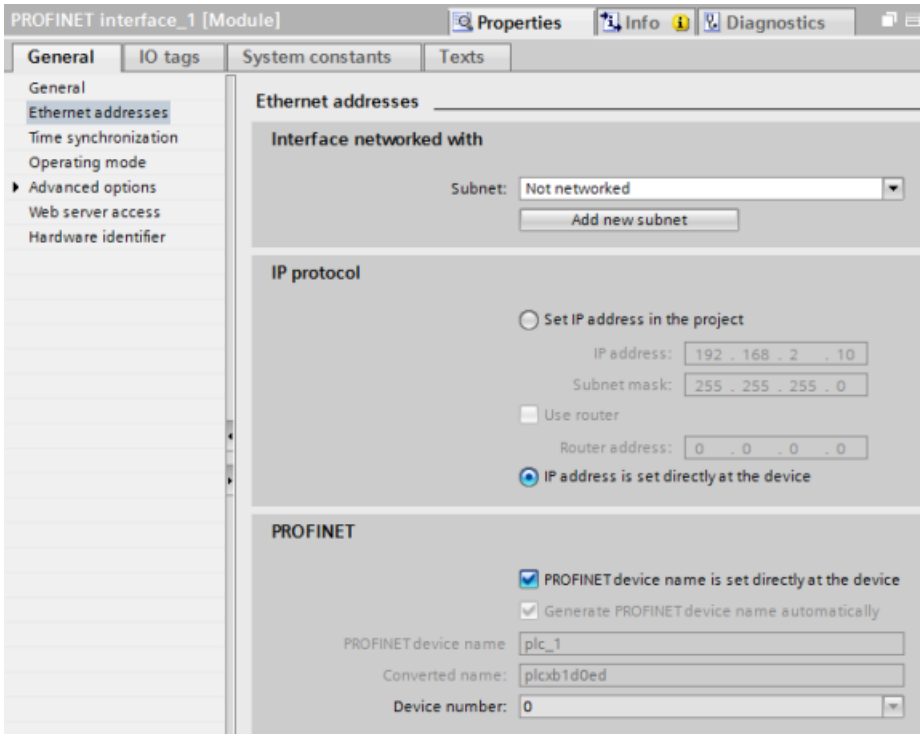

PROFINET interface_1 [Module]
Properties
Info
Diagnostics
General
IO tags
System constants
Texts
General
Ethernet addresses
Time synchronization
Operating mode
Advanced options
Web server access
Hardware identifier
Ethernet addresses
Interface networked with
Subnet:
Not networked
Add new subnet
IP protocol
Set IP address in the project
IP address:
192 . 168 . 2 . 10
Subnet mask:
255 . 255 . 255 . 0
Use router
Router address:
0 . 0 . 0 . 0
IP address is set directly at the device
PROFINET
PROFINET device name is set directly at the device
Generate PROFINET device name automatically
PROFINET device name
plc_1
Converted name:
plcxb1d0ed
Device number:
0

Пример: Изменение параметров IP и имен устройств PROFINET IO с помощью инструкции T_CONFIG

В следующем примере оба субполя "addr" и "nos" (Name of station) будут изменены. На странице "Ethernet адрес" в "Свойствах" CPU с помощью радиокнопки "Имя устройства PROFINET устанавливается непосредственно на устройстве" после загрузки программы в CPU можно изменить имя устройства PROFINET через инструкцию "T_CONFIG".

CONF_DATA_2

	Name	Data type	Start value
1	Static		
2	Conf_data	Struct	
3	header	IF_CONF_Header	
4	FieldType	UInt	0
5	FieldId	UInt	0
6	SubfieldCount	UInt	2
7	addr	IF_CONF_v4	
8	Id	UInt	30
9	Length	UInt	18
10	Mode	UInt	1
11	InterfaceAddress	IP_V4	
12	ADDR	array [1..4] of Byte	
13	SubnetMask	IP_V4	
14	ADDR	array [1..4] of Byte	
15	DefaultRouter	IP_V4	
16	ADDR	array [1..4] of Byte	
17	nos	IF_CONF_NOS	
18	Id	UInt	40
19	Length	UInt	246
20	Mode	UInt	1
21	NOS	array [1..240] of Byte	



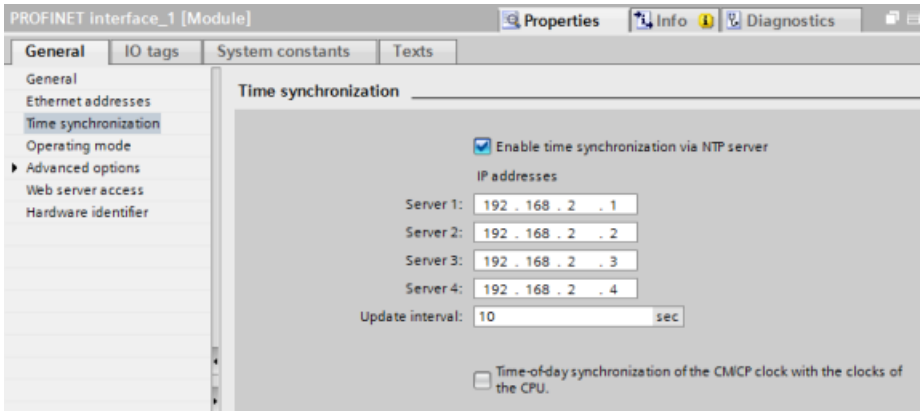
Пример: Изменение IP-адресов на NTP-серверах через инструкцию T_CONFIG

В следующем примере инструкция T_CONFIG в субполе "ntp" (Network-Time-Protocol-Server (NTP)) изменяет IP-адреса до четырех NTP-серверов.

На странице "Синхронизация времени" в "Свойствах" CPU, PROFINET интерфейс [X1], NTP-синхронизация настраивается путем установки флажка "Активировать синхронизацию времени через NTP-сервер" (см. рисунок ниже). После загрузки программы в CPU можно изменять IP-адрес NTP-сервера при помощи инструкции "T_CONFIG".

CONF_DATA_3

	Name	Data type	Start value
1	Static		
2	Conf_Data	Struct	
3	header	IF_CONF_Header	
4	FieldType	UInt	0
5	FieldId	UInt	0
6	SubfieldCount	UInt	1
7	ntp	IF_CONF_NTP	
8	Id	UInt	17
9	Length	UInt	22
10	Mode	UInt	2
11	NTP_IP	Array[1..4] of IP_V4	
12	NTP_IP[1]	IP_V4	
13	ADDR	Array[1..4] of Byte	
14	ADDR[1]	Byte	192
15	ADDR[2]	Byte	168
16	ADDR[3]	Byte	2
17	ADDR[4]	Byte	5
18	NTP_IP[2]	IP_V4	
19	ADDR	Array[1..4] of Byte	
20	ADDR[1]	Byte	192
21	ADDR[2]	Byte	168
22	ADDR[3]	Byte	2
23	ADDR[4]	Byte	6
24	NTP_IP[3]	IP_V4	
25	ADDR	Array[1..4] of Byte	
26	ADDR[1]	Byte	192
27	ADDR[2]	Byte	168
28	ADDR[3]	Byte	2
29	ADDR[4]	Byte	7
30	NTP_IP[4]	IP_V4	
31	ADDR	Array[1..4] of Byte	
32	ADDR[1]	Byte	192
33	ADDR[2]	Byte	168
34	ADDR[3]	Byte	2
35	ADDR[4]	Byte	8



11.2.8.18 Общие параметры для инструкций

Входной параметр REQ

Многие инструкции для открытых коммуникационных соединений пользователя используют вход REQ для запуска операции по положительному фронту (0 на 1). Вход REQ должен быть 1 (TRUE) при выполнении инструкции, но вход REQ может оставаться TRUE столько времени, сколько необходимо. Инструкция не инициирует другую операцию, пока она не будет выполнена с входом REQ = FALSE, чтобы команда могла сбросить состояние входа REQ. Это необходимо для того, чтобы инструкция могла обнаружить положительный фронт для запуска следующей операции.

В случае добавления в программу одной из таких инструкции, STEP 7 потребует указать экземплярный блок данных. Для каждого вызова инструкции следует использовать уникальный блок данных. Это гарантирует, что такие входы как REQ будут правильно обрабатываться каждой инструкцией.

Входной параметр ID

Это ссылка на "Локальный ID (шестн.)" в "Просмотре сетевых соединений" в " Устройства и сети" в STEP 7 и это идентификатор сети, которая будет использоваться для этого коммуникационного блока. ID должен совпадать с соответствующим ID параметра в локальном описании соединения.

Выходные параметры DONE, NDR, ERROR и STATUS

У этих инструкций есть выходы, отображающие состояние выполнения:

Таблица 11-55 Выходные параметры инструкций для открытых коммуникационных соединений пользователя

Параметр	Тип данных	По умолчанию	Описание
DONE	Bool	FALSE	Устанавливается на TRUE на один цикл выполнения, чтобы показать, что последний запрос был завершен с ошибками; в противном случае FALSE.
NDR	Bool	FALSE	Устанавливается на TRUE на один цикл выполнения, чтобы показать, что запрошенная операция была завершена без ошибок и были получены новые данные; в противном случае FALSE.
BUSY	Bool	FALSE	При активности устанавливается на TRUE, чтобы показать следующее: • Операция еще не завершена. • Новая операция не может быть запушена. Устанавливается на FALSE, если операция завешена.

Параметр	Тип данных	По умолчанию	Описание
ERROR	Bool	FALSE	Устанавливается для одного цикла выполнения на TRUE, чтобы показать, что последний запрос завершен с ошибками, соответствующий код ошибки находится в STATUS; в противном случае FALSE.
STATUS	Word	0	Статус результата: • Если устанавливается бит DONE или NDR, то STATUS устанавливается на 0 или на информационный код. • Если устанавливается бит ERROR, то STATUS устанавливается на код ошибки. • Если не устанавливается ни один из вышеуказанных битов, инструкция может вернуть результаты состояния, которые описывают текущее состояние функции. Значение STATUS сохраняется при выполнении функции.

Примечание

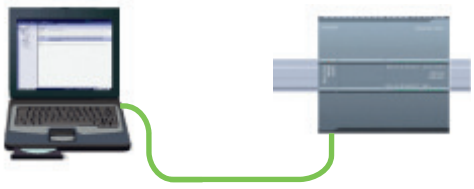
Следует учитывать, что DONE, NDR и ERROR остаются установленными только в течение одного цикла выполнения.

Запрещенные номера TSAP и портов для пассивной ISO и TCP коммуникации

При использовании инструкции TCON для настройки и установки пассивного коммуникационного соединения, следующие адреса портов запрещены:

- ISO TSAP (пассивная):
  - 01.00, 01.01, 02.00, 02.01, 03.00, 03.01
  - 10.00, 10.01, 11.00, 11.01, ... BF0,00, BF0,01
- Порт TCP (пассивный): 5001, 102, 123, 20, 21, 25, 34962, 34963, 34964, 80
- Порт UDP (пассивный): 161, 34962, 34963, 34964

11.2.9 Обмен данными с программатором



CPU может обмениваться данными с находящимся в сети программатором со STEP 7.

При настройке коммуникаций между CPU и программатором необходимо принять во внимание следующее:

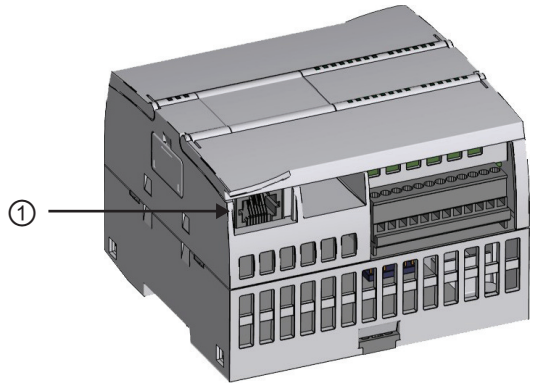
- Конфигурация/установка: Требуется аппаратное конфигурирование.
- Для коммуникации между двумя устройствами Ethernet коммутатор не нужен; только при наличии более двух устройств в одной сети потребуется Ethernet коммутатор.

11.2.9.1 Установление аппаратного коммуникационного соединения

Интерфейс PROFINET является физическим соединением между программатором и CPU. Поскольку функция автоматического определения типа кабеля встроена в CPU, для интерфейса можно использовать стандартный Ethernet кабель или Ethernet кабель с перекрестными соединениями. Коммутатор Ethernet не требуется для прямого подключения программатора к CPU.

Для создания аппаратного соединения между программатором и CPU действовать следующим образом:

1. Установить CPU (Страница 60).
2. Вставьте кабель Ethernet в порт PROFINET, как показано ниже.
3. Подключите кабель Ethernet к программатору.



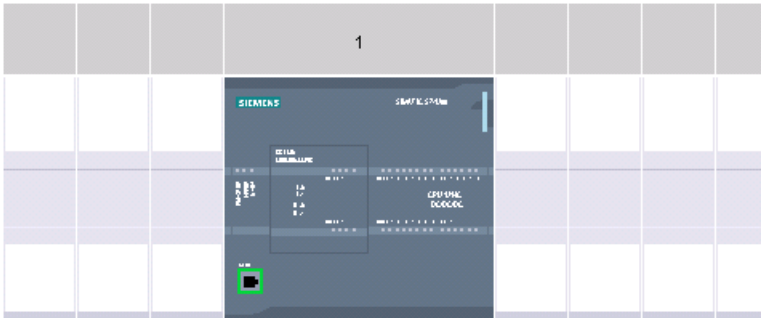
① Порт PROFINET

Для подключения PROFINET имеется опциональный компенсатор натяжения кабеля. Для получения информации для заказа см. раздел Запасные части и другое оборудование (Страница 1642).

11.2.9.2 Конфигурирование устройств

Если проект с CPU уже был создан, открыть его в STEP 7.

Если нет, создать проект и вставить CPU (Страница 152) в стойку. В следующем проекте CPU показан в "Просмотре устройств".



11.2.9.3 Назначение IP-адресов

Назначение IP-адресов

В сети PROFINET каждое устройство должно также иметь IP-адрес (адрес протокола Интернет). Этот адрес позволяет устройству передавать данные по более сложной, маршрутизированной сети:

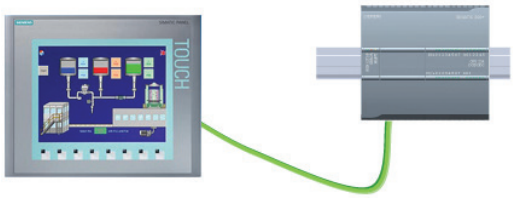
- При использовании программатора или других сетевых устройств со встроенной адаптерной картой, подключенной к локальной сети предприятия, или адаптерной картой Ethernet-USB с подключением к изолированной сети, необходимо назначить этим устройствам IP-адреса. Обратиться к разделу "Назначение IP-адресов для программаторов и сетевых устройств" (Страница 815) для получения дополнительной информации.
- Можно назначить IP-адрес CPU или сетевому устройству и в режиме онлайн. Это особенно полезно при первом конфигурировании устройства. Дополнительную информацию можно найти здесь "Назначение IP-адреса CPU в режиме онлайн (Страница 815)".
- После завершения конфигурирования CPU или сетевого устройства в проекте, можно настроить параметры для PROFINET интерфейса, чтобы записать IP-адрес. Дополнительную информацию можно найти здесь "Конфигурирование IP-адреса для CPU в проекте" (Страница 817).

11.2.9.4 Тестирование сети PROFINET

После завершения конфигурирования необходимо загрузить проект в CPU. Все IP-адреса конфигурируются при загрузке проекта.

Функция CPU "Загрузка в устройство" и ее диалоговое окно "Расширенная загрузка" могут показать все доступные сетевые устройства, а также всем ли устройствам назначены уникальные IP-адреса. Дополнительную информацию см. "Тестирование сети PROFINET" (Страница 823).

11.2.10 Коммуникация HMI/PLC



CPU поддерживает два коммуникационных соединения PROFINET с HMI (Страница 32). Для установки связи между CPU и HMI должны быть выполнены следующие условия:

Конфигурация/установка:

- PROFINET порт CPU должен быть сконфигурирован для соединения с HMI.
- HMI должен быть настроен и сконфигурирован.
- Данные конфигурации HMI являются частью проекта CPU и могут быть настроены и загружены в проекте.
- Для коммуникации между двумя устройствами Ethernet коммутатор не нужен; только при наличии более двух устройств в одной сети потребуется Ethernet коммутатор.

Примечание

Также можно использовать монтируемый в стойку Ethernet коммутатор CSM1277 с 4 портами для подключения нескольких CPU и устройств HMI. Порт PROFINET на CPU не содержит Ethernet коммутатора.

Поддерживаемые функции:

- HMI может читать /записывать данные CPU.
- Сообщения могут запускаться на основании информации от CPU.
- Диагностика системы

Таблица 11-56 Необходимые шаги по конфигурированию соединения между HMI и CPU

Шаг	Постановка задачи
1	Установление аппаратного коммуникационного соединения Интерфейс PROFINET является физическим соединением между HMI и CPU. Поскольку функция автоматического определения типа кабеля встроена в CPU, для интерфейса можно использовать стандартный Ethernet кабель или Ethernet кабель с перекрестными соединениями. Коммутатор Ethernet не требуется для прямого подключения HMI к CPU. Дополнительную информацию можно найти здесь: "Обмен данными с программатором: установление аппаратного коммуникационного соединения (Страница 932)".
2	Конфигурирование устройств Дополнительную информацию можно найти здесь: "Обмен данными с программатором: конфигурирование устройств (Страница 932)".
3	Конфигурирование логических сетевых соединений между HMI и CPU Дополнительную информацию можно найти здесь: "Коммуникация между HMI и PLC: конфигурирование логических сетевых соединений между двумя устройствами" (Страница 935).

Шаг	Постановка задачи
4	Конфигурирование IP-адреса в проекте Действовать схожим образом. Но необходимо сконфигурировать IP-адреса для HMI и CPU. Дополнительную информацию см. в "Конфигурация устройства: конфигурирование IP-адреса для CPU в проекте" (Страница 819).
5	Тестирование сети PROFINET Необходимо загрузить конфигурацию для каждого CPU и каждого устройства HMI. Подробную информацию см. в "Конфигурация устройства: тестирование сети PROFINET" (Страница 823).

11.2.10.1 Конфигурирование логических сетевых соединений между двумя устройствами

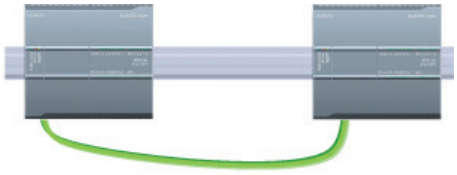
После конфигурирования стойки с CPU, можно сконфигурировать сетевые соединения.

На портале "Устройства и сети" в "Просмотре сетевых соединений" можно объединять в сеть устройства в проекте. Сначала следует открыть вкладку "Соединения" и выбрать в выпадающем списке справа тип соединения (например, соединение ISO-on-TCP).

Для создания соединения PROFINET, кликнут по зеленому полю (PROFINET) на первом устройстве и провести линию к полю PROFINET на втором устройстве. Отпустить кнопку мыши. Соединение PROFINET создано.

Подробную информацию см. в "Конфигурация устройства: создание сетевого соединения" (Страница 811).

11.2.11 Коммуникация PLC/PLC



Один CPU с помощью инструкций TSEND_C и TRCV_C может обмениваться данными с другим CPU по сети.

При настройке коммуникаций между двумя CPU необходимо принять во внимание следующее:

- Конфигурация/установка: Требуется аппаратное конфигурирование.
- Поддерживаемые функции: Чтение/запись данных в сетевом CPU
- Для коммуникации между двумя устройствами Ethernet коммутатор не нужен; только при наличии более двух устройств в одной сети потребуется Ethernet коммутатор.

Таблица 11-57 Необходимые шаги по конфигурированию соединения между двумя CPU

Шаг	Постановка задачи
1	Установление аппаратного коммуникационного соединения Интерфейс PROFINET является физическим соединением между двумя CPU. Поскольку функция автоматического определения типа кабеля встроена в CPU, для интерфейса можно использовать стандартный Ethernet кабель или Ethernet кабель с перекрестными соединениями. Коммутатор Ethernet не требуется для соединения двух CPU. Дополнительную информацию можно найти здесь: "Обмен данными с программатором: установление аппаратного коммуникационного соединения (Страница 932)".
2	Конфигурирование устройств Необходимо сконфигурировать оба CPU в проекте. Дополнительную информацию можно найти здесь: "Обмен данными с программатором: конфигурирование устройств (Страница 932)".
3	Конфигурирование логических сетевых соединений между двумя CPU Дополнительную информацию можно найти здесь: "Коммуникация между двумя PLC: конфигурирование логических сетевых соединений между двумя устройствами" (Страница 936).
4	Конфигурирование IP-адреса в проекте Действовать схожим образом. Но необходимо сконфигурировать IP-адреса для двух CPU (напр., PLC_1 и PLC_2). Дополнительную информацию см. в "Конфигурация устройства: конфигурирование IP-адреса для CPU в проекте" (Страница 819).
5	Конфигурирование параметров передачи и приема Необходимо сконфигурировать инструкции TSEND_C и TRCV_C в обоих CPU, чтобы обеспечить обмен данными между ними. Дополнительную информацию можно найти здесь: "Настройка коммуникации между двумя CPU: конфигурирование параметров передачи и приема (Страница 937)".
6	Тестирование сети PROFINET Необходимо загрузить конфигурацию для каждого CPU. Подробную информацию см. в "Конфигурация устройства: тестирование сети PROFINET" (Страница 823).

11.2.11.1 Конфигурирование логических сетевых соединений между двумя устройствами

После конфигурирования стойки с CPU, можно сконфигурировать сетевые соединения.

На портале "Устройства и сети" в "Просмотре сетевых соединений" можно объединять в сеть устройства в проекте. Сначала следует открыть вкладку "Соединения" и выбрать в выпадающем списке справа тип соединения (например, соединение ISO-on-TCP).

Для создания соединения PROFINET, кликнут по зеленому полю (PROFINET) на первом устройстве и провести линию к полю PROFINET на втором устройстве. Отпустить кнопку мыши. Соединение PROFINET создано.

Подробную информацию см. в "Конфигурация устройства: создание сетевого соединения" (Страница 811).

11.2.11.2 Конфигурирование пути соединения между локальным и партнерским CPU

Конфигурирование общих параметров

Параметры коммуникации определяются в диалоговом окне "Свойства" коммуникационной инструкции. Это диалоговое окно отображается в нижней части страницы каждый раз, когда выбирается любая часть инструкции.

Дополнительную информацию см. в "Конфигурация устройства: конфигурирование пути соединения между локальным и партнерским CPU (Страница 812)".

Используемые TSAP или порты определяются в диалоговом окне "Параметры соединения" в пункте "Адресная информация". TSAP или порт соединения в CPU вводится в поле "Локальный TSAP". TSAP или порт для соединения в партнерском CPU вводится в поле "Партнерская TSAP".

11.2.11.3 Конфигурирование параметров передачи и приема

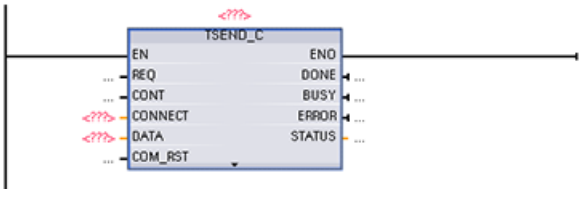
Коммуникационные блоки (например, TSEND_C и TRCV_C) используются, чтобы установить соединения между двумя CPU. Прежде чем CPU смогут участвовать в PROFINET-коммуникациях, необходимо сконфигурировать параметры для передачи и приема сообщений. Эти параметры определяют, как работает коммуникация, когда сообщения передаются на / принимаются от целевого устройства.

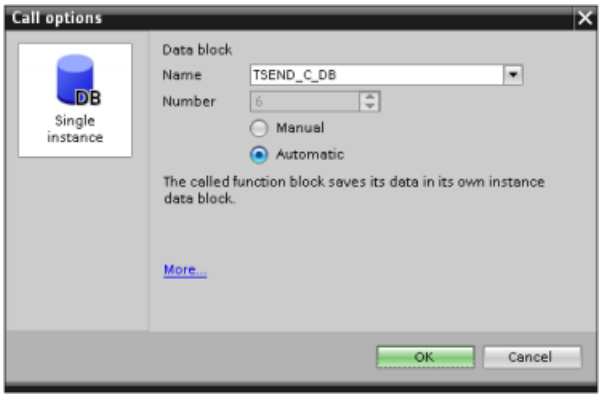
Конфигурирование параметров передачи для TSEND_C

Инструкция TSEND_C

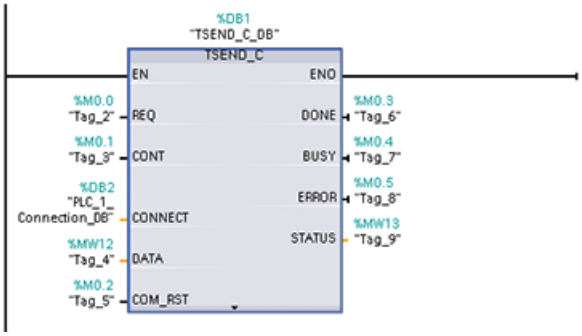
Инструкция TSEND_C (Страница 843) создает коммуникационное соединение со станцией-партнером. Соединение настраивается, устанавливается и автоматически контролируется, пока не поступит команда на разъединение от инструкции. Инструкция TSEND_C комбинирует функции инструкций TCON, TDISCON и TSEND.

В STEP 7 в конфигурации устройства можно установить, как данные должны передаваться с помощью инструкции TSEND_C. Сначала необходимо вставить инструкцию из папки "Коммуникация" в окне задач "Инструкции" в программу. Инструкция TSEND_C отображается вместе с диалоговым окном "Опции вызова", где может быть назначен DB для хранения параметров инструкции.





Входам и выходам может назначаться место в тегированной памяти. Это показано на рисунке ниже:



Конфигурирование общих параметров

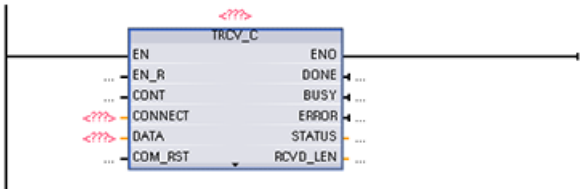
Можно определить параметры в диалоговом окне "Свойства" инструкции TSEND_C. Это диалоговое окно отображается в нижней части страницы каждый раз, когда выбирается любая часть инструкции TSEND_C.

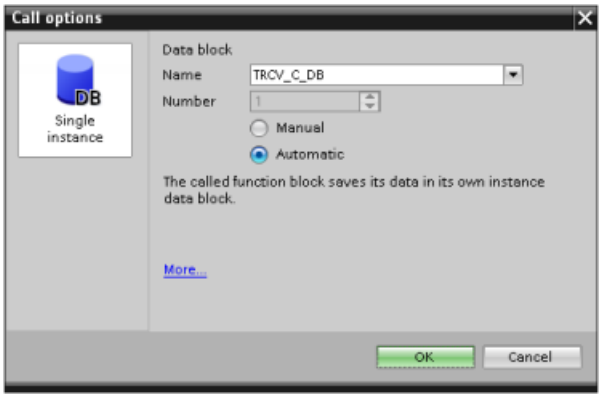
Конфигурирование параметров приема для TRCV_C

Инструкция TRCV_C

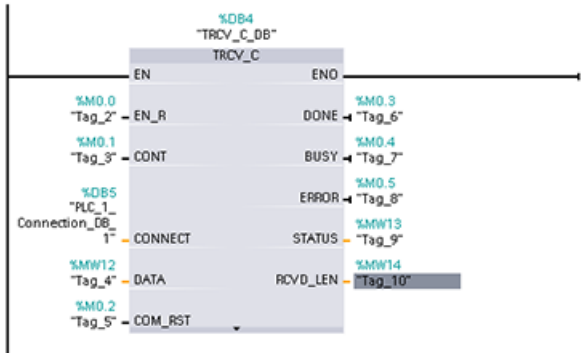
Инструкция TRCV_C (Страница 843) создает коммуникационное соединение со станцией-партнером. Соединение настраивается, устанавливается и автоматически контролируется, пока не поступит команда на разъединение от инструкции. Инструкция TRCV_C комбинирует функции инструкций TCON, TDISCON и TRCV.

В STEP 7 в конфигурации CPU можно установить, как данные должны приниматься с помощью инструкции TRCV_C. Сначала необходимо вставить инструкцию из папки "Коммуникация" в окне задач "Инструкции" в программу. Инструкция TRCV_C отображается вместе с диалоговым окном "Опции вызова", где может быть назначен DB для хранения параметров инструкции.





Входам и выходам может назначаться место в тегированной памяти. Это показано на рисунке ниже:



Конфигурирование общих параметров

Можно определить параметры в диалоговом окне "Свойства" инструкции TRCV_C. Это диалоговое окно отображается в нижней части страницы каждый раз, когда выбирается любая часть инструкции TRCV_C.

11.2.12 Конфигурирование CPU и устройства PROFINET IO

11.2.12.1 Добавление устройства PROFINET IO

Добавление PROFINET IO-устройства

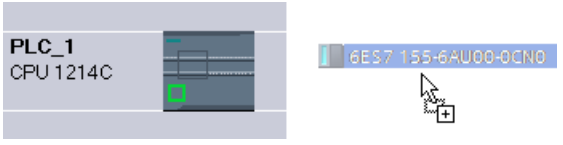
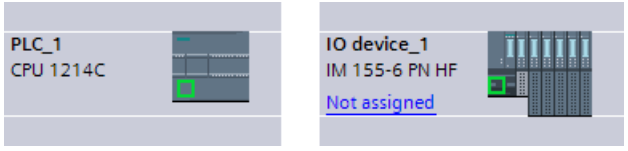
Следует использовать каталог оборудования на портале "Устройства и сети" для добавления устройств PROFINET IO.

Примечание

Для добавления устройства PROFINET IO можно использовать программу STEP 7 Professional или Basic от версии V11.

Например, для добавления ET200SP IO-устройства необходимо открыть следующие контейнеры в каталоге оборудования: Распределенная периферия, ET 200SP, интерфейсные модули и PROFINET. После можно выбрать интерфейсный модуль в списке устройств ET200SP (с сортировкой по заказным номерам) и добавить ET200SP IO-устройство.

Таблица 11-58 Добавление ET200SP IO-устройства в конфигурацию устройства

Вставка IO-устройства	Результат
	

Теперь может быть установлена связь PROFINET IO-устройства с CPU.

1. Кликнуть правой кнопкой мыши по ссылке "Не назначен" на устройстве и выбрать в контекстном меню команду "Назначить новый IO-контроллер" для вызова диалогового окна "Выбора IO-контроллера".
2. Выбрать с списке IO-контроллеров в проекте свой S7-1200 CPU (в этом примере "PLC_1").
3. Нажать кнопку "OK" для создания сетевого соединения.

Также можно перейти на портал "Устройства и сети" и там в "Просмотре сетевых соединений" объединять в сеть устройства в проекте:

1. Для создания соединения PROFINET, кликнут по зеленому полю (PROFINET) на первом устройстве и провести линию к полю PROFINET на втором устройстве.
2. Отпустить кнопку мыши. Соединение PROFINET создано.

Дополнительную информацию см. в Конфигурация устройства: конфигурирование CPU для коммуникации" (Страница 175).

11.2.12.2 Присвоение имен CPU и устройствам

Присвоение имен CPU и устройствам

Сетевые соединения между устройствами также назначают PROFINET IO устройство CPU для того, чтобы CPU мог управлять устройством. Чтобы изменить это назначение, кликнуть по имени PLC, отображаемому на PROFINET IO-устройстве. Открывается диалоговое окно, которое позволяет отключить PROFINET IO-устройство от текущего CPU и переназначить его или оставить, при необходимости, не назначенным.

У устройств в PROFINET-сети должно быть назначенное имя, прежде чем они смогут подключаться к CPU. На вкладке просмотра сетевых соединений PROFINET-устройствам присваиваются имена, если у устройств еще нет имен или если имя устройства должно быть изменено. Кликнуть правой кнопкой по PROFINET IO устройству и выбрать "Присвоить имя устройства".

Имя PROFINET IO-устройства должно быть присвоено как в проекте STEP 7, так и PROFINET IO-устройству в сети PROFINET. (Для присвоения имени устройства в сети PROFINET можно использовать либо инструмент STEP 7 "Онлайн и диагностика", либо инструмент PRONETA для ввода в эксплуатацию, конфигурирования и диагностики.) При отсутствии имени или несовпадении имен в обоих местах хранения, режим для обмена данными PROFINET IO не будет работать. Дополнительную информацию можно найти здесь "Интерактивные и диагностические инструменты: присвоение имени устройству PROFINET в режиме онлайн (Страница 1391)".

11.2.12.3 Назначение IP-адресов

Назначение IP-адресов

В сети PROFINET каждое устройство должно также иметь IP-адрес (адрес протокола Интернет). Этот адрес позволяет устройству передавать данные по более сложной, маршрутизированной сети:

- При использовании программатора или других сетевых устройств со встроенной адаптерной картой, подключенной к локальной сети предприятия, или адаптерной картой Ethernet-USB с подключением к изолированной сети, необходимо назначить этим устройствам IP-адреса. Обратиться к разделу "Назначение IP-адресов для программаторов и сетевых устройств" (Страница 815) для получения дополнительной информации.
- Можно назначить IP-адрес CPU или сетевому устройству и в режиме онлайн. Это особенно полезно при первом конфигурировании устройства. Дополнительную информацию можно найти здесь "Назначение IP-адреса CPU в режиме онлайн (Страница 817)".
- После завершения конфигурирования CPU или сетевого устройства в проекте, можно настроить параметры для PROFINET интерфейса, чтобы записать IP-адрес. Дополнительную информацию можно найти здесь "Конфигурирование IP-адреса для CPU в проекте" (Страница 819).

11.2.12.4 Конфигурирование времени цикла IO

Конфигурирование времени цикла IO

PROFINET IO-устройство получает новые данные от CPU в течение промежутка времени "IO-цикл". Время обновления может быть настроено отдельно для каждого устройства, и оно определяет интервал времени для передачи данных между CPU и устройством.

STEP 7 рассчитывает время обновления для "IO цикла" автоматически в настройке по умолчанию для каждого устройства сети PROFINET, принимая во внимание объем данных для передачи, и число устройств, назначенных этому контроллеру. Если автоматическое вычисление времени обновления не требуется, то можно изменить эти настройки.

Параметры для "IO цикла" определяются в конфигурационном диалоге "Свойства" PROFINET IO устройства. Это диалоговое окно отображается в нижней части страницы каждый раз, когда выбирается любая часть инструкции.

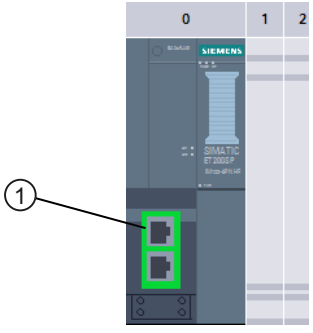
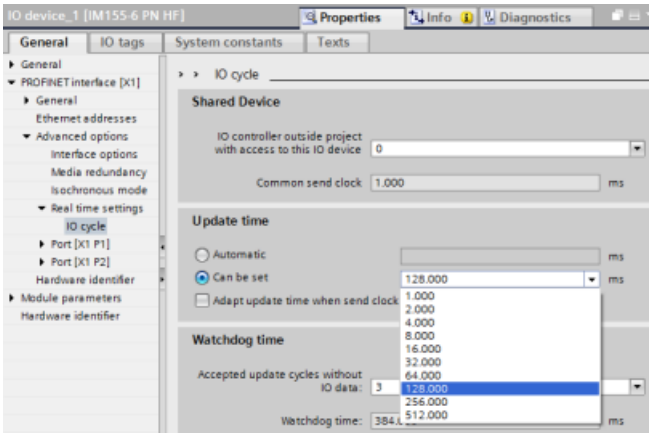
В представлении "Просмотр устройств" для PROFINET IO устройства кликнуть по порту PROFINET. В диалоговом окне "Интерфейс PROFINET" вызвать параметры "IO цикла", выбирая следующие пункты меню:

- "Расширенные опции"
- "Настройки реального времени"
- "IO-цикл"

Определить "время обновления" для IO цикла с помощью следующих настроек:

- Чтобы получить надлежащее время обновления, вычисленное автоматически, выбрать "Автоматическое".
- Чтобы выбрать время обновления самостоятельно, выбрать пункт "Настраиваемое" и указать нужное время обновления в мс.

Таблица 11-59 Конфигурирование времени цикла PROFINET IO для ET200SP

ET 200SP PROFINET IO-устройство	Диалоговое окно "ET 200SP PROFINET IO-цикл"
	

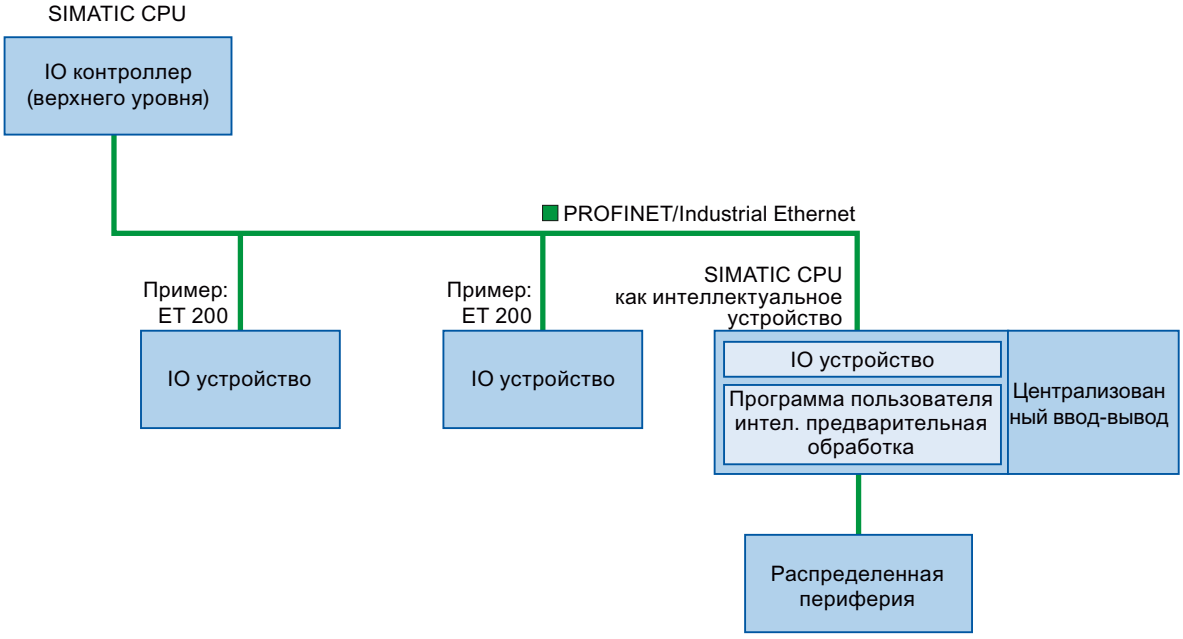
① Порт PROFINET

11.2.13 Конфигурирование CPU и интеллектуального устройства PROFINET

11.2.13.1 Функциональные возможности I устройства

Функция "I-устройство" (интеллектуальное IO устройство) CPU упрощает обмен данными с IO контроллером и использование CPU в качестве, например, интеллектуального модуля предварительной обработки отдельных фаз технологического процесса. I устройство подключается как IO устройство к IO контроллеру более высокого уровня.

Предварительная обработка осуществляется через программу пользователя CPU. Значения процесса, собранные при централизованном или распределенном (PROFINET IO или PROFIBUS DP) вводе-выводе, предварительно обрабатываются пользовательской программой и становятся доступными посредством PROFINET IO интерфейса для CPU станции более высокого уровня.



Правило создания имен для "I устройства"

В дальнейшем CPU или CP с функциональностью I-устройства называется просто "I-устройство".

11.2.13.2 Свойства и преимущества I-устройства

Области применения

Области применения для I-устройства:

- Распределенная обработка:
  Сложные задачи автоматизации могут быть разбиты на небольшие блоки / подпроцессы. Это делает процессы лучше управляемыми и упрощает подзадачи.
- Выделение подпроцессов:
  Сложные и масштабные распределенные процессы могут быть разделены на несколько подпроцессов с удобными в управлении интерфейсами с помощью I-устройств. Эти подпроцессы при необходимости могут быть сохранены в отдельных проектах STEP 7, которые позже могут быть объединены в один основной проект.
- Защита ноу-хау:
  Компоненты могут быть поставлены в виде единственного GSD-файла для описания интерфейсов I-устройства вместо проекта STEP 7. Пользователь может защитить свою программу, так как ее публикация больше не требуется.

Свойства

Свойства I-устройства:

- Независимость от проектов STEP 7:
  Создатели и пользователи I-устройств могут иметь полностью независимые проекты автоматизации STEP 7. Интерфейс между проектами STEP 7 создает файл GSD. Это позволяет организовать канал связи со стандартными IO контроллерами через стандартизированный интерфейс.
- Коммуникация в режиме реального времени:
  I-устройство поддерживает детерминированную систему PROFINET IO через интерфейс PROFINET IO.

Преимущества

I-устройство обладает следующими преимуществами:

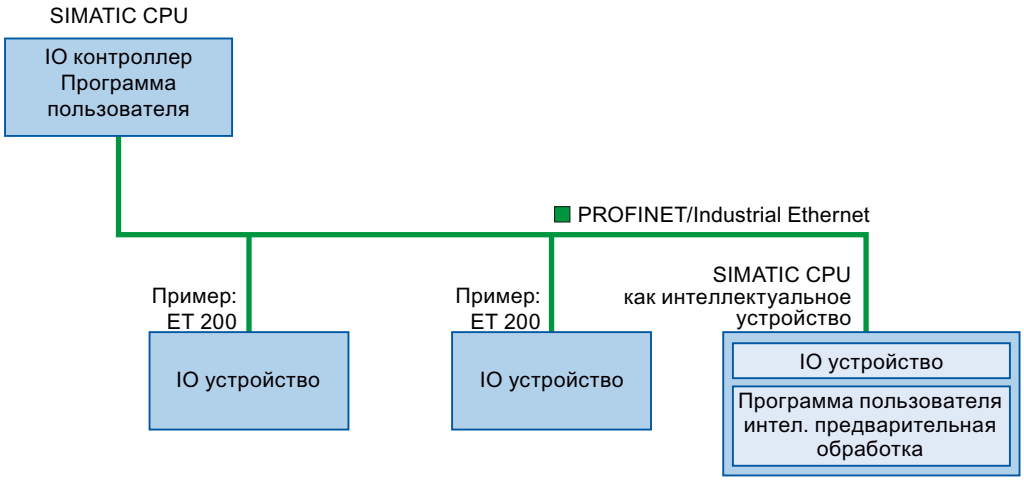
- Простое подключение к IO-контроллерам
- Коммуникации в режиме реального времени между IO-контроллерами
- Разгрузка IO-контроллера путем распределения вычислительной нагрузки на I-устройства
- Снижение коммуникационной нагрузки благодаря локальной обработке данных процесса
- Более удобные в управлении проекты за счет обработки подзадач в отдельных проектах STEP 7

11.2.13.3 Отличительные особенности I-устройства

I-устройство встраивается в систему ввода-вывода как стандартное IO устройство.

I-устройство без системы PROFINET IO более низкого уровня

У I-устройства нет собственной распределенной периферии. Конфигурирование и назначение параметров I-устройства как IO-устройства идентичны таковым для распределенной системы ввода-вывода (например, ET 200).



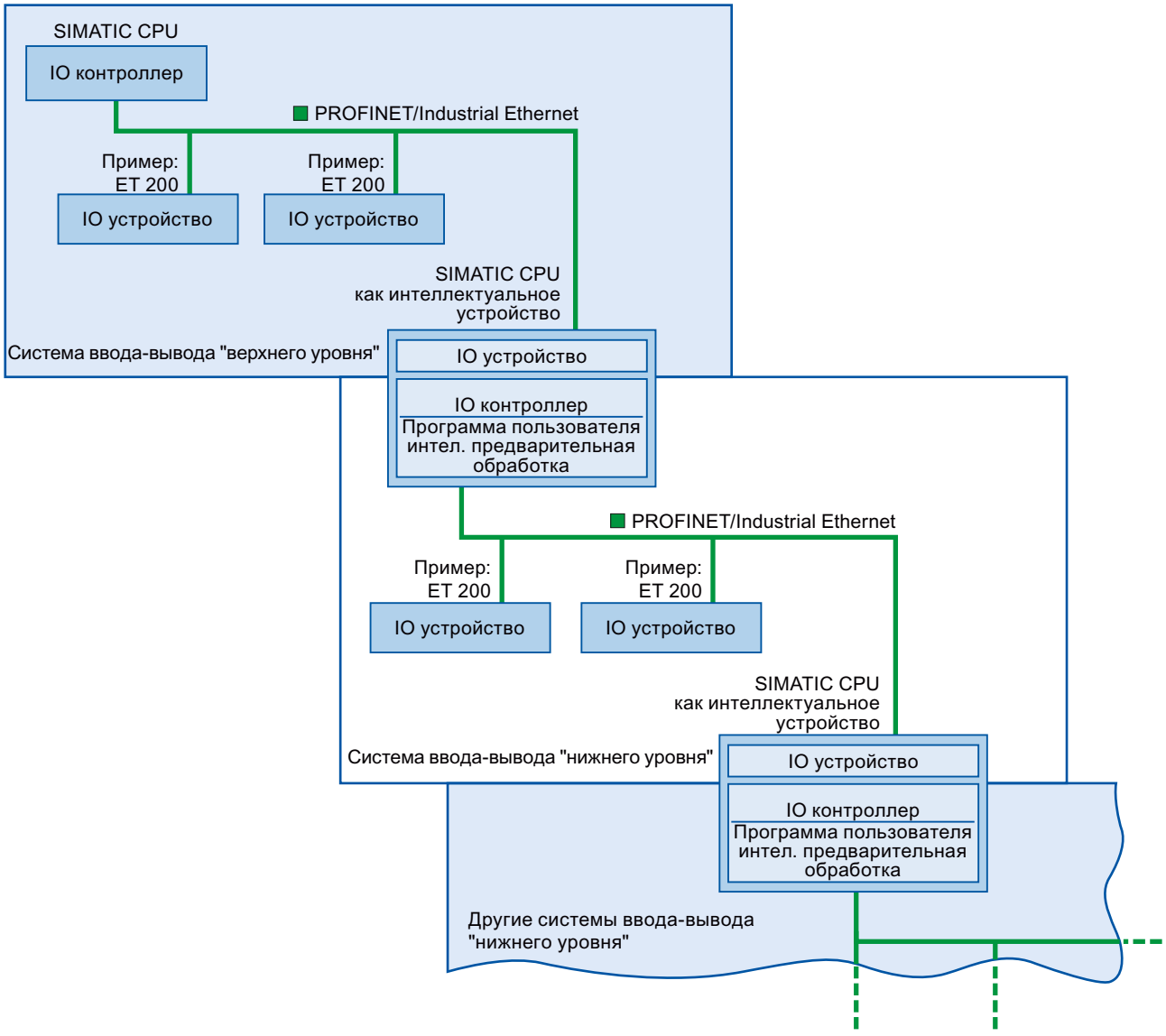
I-устройство с системой PROFINET IO более низкого уровня

В зависимости от конфигурации, I-устройство может также быть IO-контроллером на PROFINET-интерфейсе в дополнение к роли IO-устройства.

Это означает, что I-устройство может быть частью IO-системы более высокого уровня посредством своего PROFINET-интерфейса, а в качестве IO-контроллера может поддерживать свою собственную IO-систему более низкого уровня.

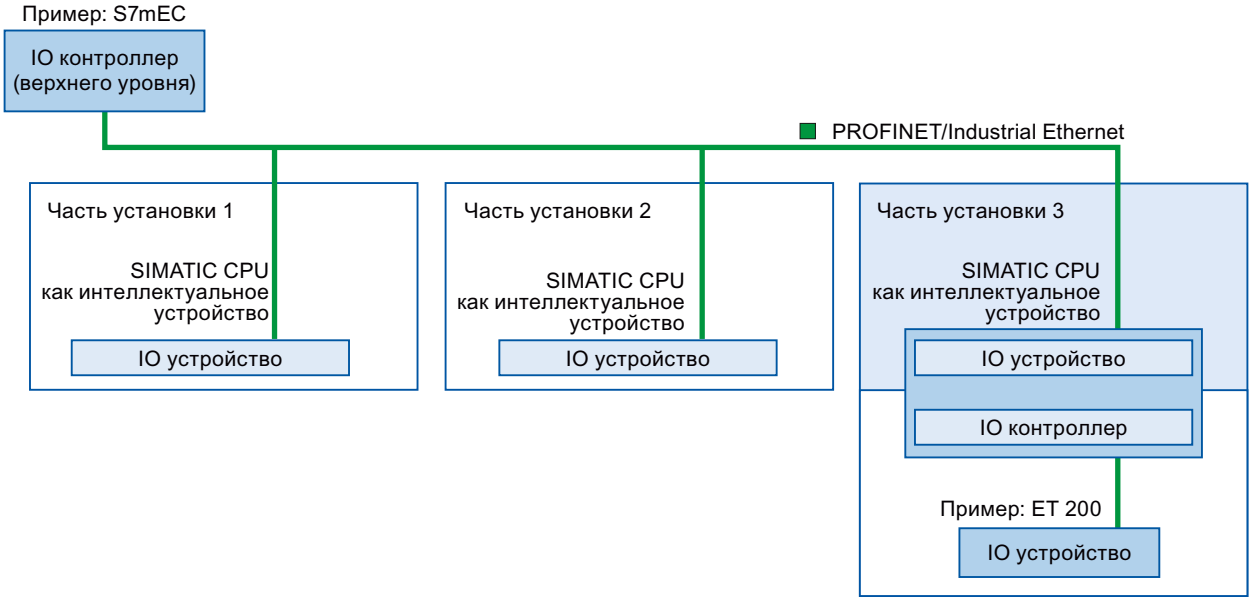
IO-система более низкого уровня может, в свою очередь, также содержать I-устройства (см. рисунок ниже). Это делает возможным построение иерархически структурированных IO-систем.

В дополнение к своей роли в качестве IO-контроллера, I-устройство может также использоваться через PROFIBUS-интерфейс в качестве ведущего устройства DP для системы PROFIBUS более низкого уровня.


SIMATIC CPU
IO контроллер
PROFINET/Industrial Ethernet
Пример:
ET 200
IO устройство
Пример:
ET 200
IO устройство
SIMATIC CPU
как интеллектуальное
устройство
Система ввода-вывода "верхнего уровня"
IO устройство
IO контроллер
Программа пользователя
интел. предварительная
обработка
PROFINET/Industrial Ethernet
Пример:
ET 200
IO устройство
Пример:
ET 200
IO устройство
SIMATIC CPU
как интеллектуальное
устройство
Система ввода-вывода "нижнего уровня"
IO устройство
IO контроллер
Программа пользователя
интел. предварительная
обработка
Другие системы ввода-вывода
"нижнего уровня"

Пример: I-устройство в качестве IO устройства и IO контроллера

Использование I-устройства в качестве IO устройства и IO контроллера здесь объясняется на примере процесса печати. I-устройство управляет отдельным блоком установки (подпроцесс). Один из нескольких блоков используется, например, чтобы вкладывать дополнительные листы, напр., флайеры или брошюры, в пакет с печатной продукцией.



Каждый блок 1 и блок 2 состоит из I-устройства с централизованным вводом-выводом. I-устройство вместе с распределенной системой ввода-вывода (например, ET 200) образуют блок 3.

Пользовательская программа в I-устройстве отвечает за предварительную обработку данных процесса. Для этой задачи программе пользователя I-устройства достаточно настроек по умолчанию (например, управляющей информации) от IO-контроллера верхнего уровня. I-устройство предоставляет IO-контроллеру верхнего уровня результаты (например, состояние его подзадачи).

11.2.13.4 Обмен данными между системами ввода-вывода верхнего и нижнего уровня

Области передачи представляют собой интерфейс к программе пользователя CPU I-устройства. Входы обрабатываются в программе пользователя, а выходы являются результатом обработки.

В областях передачи находятся данные для коммуникации между IO контроллером и I-устройством. Область передачи содержит информационный блок, которым последовательно обмениваются IO контроллер и I-устройство. Больше информации о конфигурации и использовании областей передачи можно найти в главе "Конфигурирование I-устройства" (Страница 950).

Различное поведение областей передачи входов у контроллера и I-устройства при потере сети

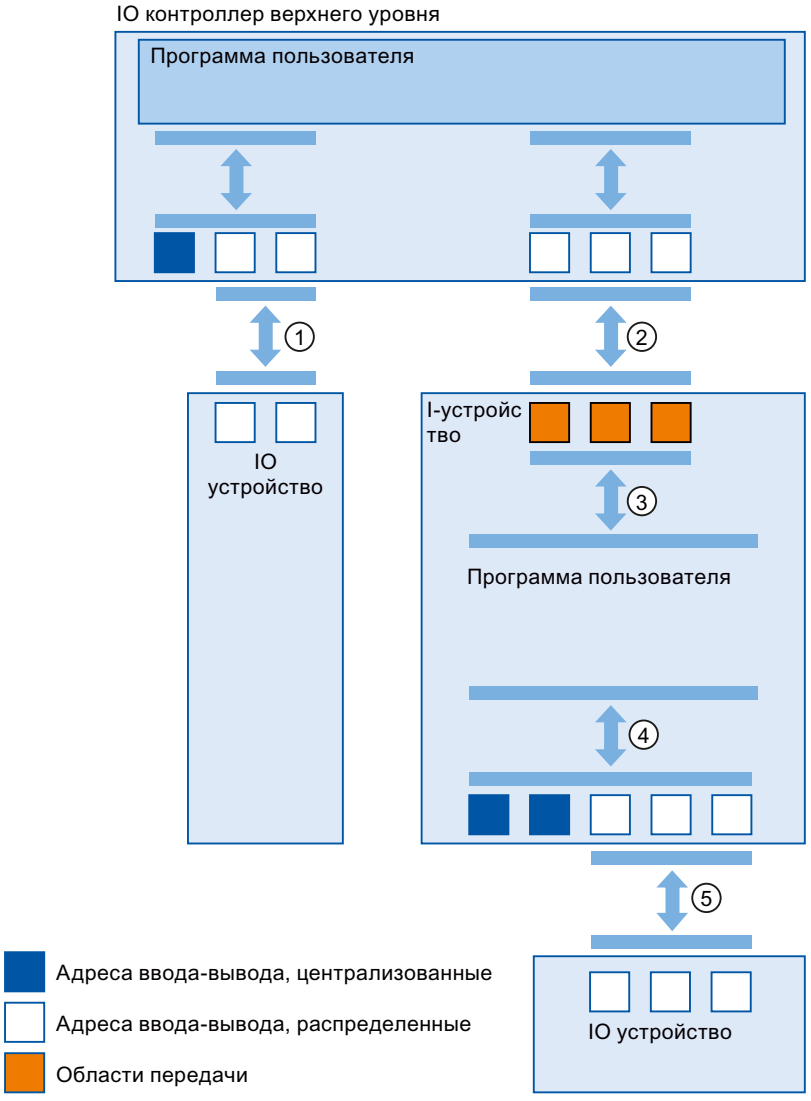
В случае контроллера CPU при потере сетевого соединения записывает ноль в области передачи входов. На I-устройстве области передачи сохраняют свои последние значения.

Путем соответствующего конфигурирования системы можно отключить такое поведение для общих случаев использования I-устройства (non-shared I-device). Для этого следует удалить области передачи входов для I-устройства в OB для отказа стойки или станции OB для наступающего события. Выполнить следующие действия:

1. Вставить OB для отказа стойки или станции в проект. (Номером этого OB по умолчанию является OB 86.)
2. Добавьте логику в OB, чтобы записать ноль для значений входов I-устройства, когда пусковая переменная LAD показывает значение аппаратного идентификатора I-устройства, а пусковая переменная Event_Class указывает на "наступающее" событие:
   - Аппаратный идентификатор I-устройства можно найти в таблице стандартных переменных (тегов) на вкладке "Системные константы". Аппаратный идентификатор относится к типу "HW_Device", а имя переменной (тега) показывает, что речь идет об I-устройстве (напр., "Local~PROFINET_interface_1~IODevice").
   - Значение "16#39" в Event_Class указывает на "наступающее" событие. Если входная переменная "Event_Class" содержит значение "16#39", то это означает, что OB для отказа стойки или станции сейчас активен (и не удален).

Обмен данными

Следующий рисунок показывает обмен данными между системами ввода-вывода верхнего и нижнего уровня. Отдельные коммуникационные связи объясняются ниже на основе номеров:



① **Обмен данными между IO-контроллером верхнего уровня и обычным IO-устройством**

IO-контроллер и IO-устройства обмениваются данными поPROFINET.

② **Обмен данными между IO-контроллером верхнего уровня и I-устройством**

IO-контроллер и I-устройство обмениваются данными по PROFINET.

Обмен данными между IO-контроллером верхнего уровня и I-устройством основан на стандартном взаимодействии между IO-контроллером и IO-устройством.

Для IO-контроллера верхнего уровня области передачи I-устройств представляют субдмодули предварительно сконфигурированной станции.

Выходные данные IO-контроллера являются входными данными I-устройства. Аналогично, входные данные IO-контроллера являются выходными данными I-устройства.

③ **Взаимодействие между пользовательской программой и областью передачи**
Пользовательская программа и область передачи обмениваются входными и выходными данными.

④ **Обмен данными между пользовательской программой и вводом-выводом I-устройства**
Пользовательская программа и централизованная / распределенная периферия обмениваются входными и выходными данными.

⑤ **Обмен данными между I-устройством и IO-устройством нижнего уровня**
I-устройство и его IO-устройства обмениваются данными. Передача данных осуществляется черезPROFINET.

11.2.13.5 Конфигурирование I-устройства

Существует две возможности для конфигурирования:

- Конфигурирование I-устройства в проекте
- Конфигурирование I-устройства, которое используется в другом проекте или в другой системе технических разработок.

STEP 7 позволяет сконфигурировать I-устройство для другого проекта или для другой системы технических разработок путем экспорта сконфигурированного I-устройства в файл GSD. После файл GSD импортируется в другие проекты или системы технических разработок, как и другие GSD-файлы. В этом файле GSD, наряду с другими данными, сохраняются и области передачи для обмена данными.

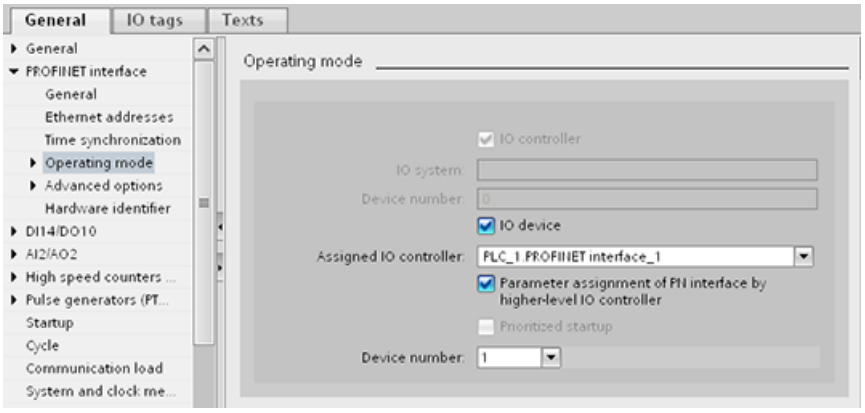
Примечание

Если S7-1200 используется как I-устройство общего доступа и как контроллер, то необходимо увеличить время обновления PROFINET I-устройства и PROFINET IO для компенсации влияния на коммуникационную способность. Система является чрезвычайно стабильной и работает правильно, если для отдельного PROFINET I-устройства и для отдельного PROFINET IO выбирается время обновления в 2 мс.

Конфигурирование I-устройства в проекте

1. Перетащить мышью PROFINET-CPU из каталога оборудования в просмотр сетевых соединений.
2. Перетащить мышью PROFINET-CPU, который также может быть сконфигурирован как IO-устройство, из каталога оборудования в просмотр сетевых соединений. Это устройство конфигурируется как I-устройство (напр., CPU 1215C).
3. Выбрать PROFINET интерфейс для I-устройства.
4. Выбрать в окне инспектора область навигации "Режим работы" и установить флажок "I-устройство".

5. Теперь в выпадающем списке "Назначенный IO-контроллер" можно выбрать IO-контроллер.
   После выбора IO-контроллера в просмотре сетевых соединений отобразится сетевое соединение и IO-система между двумя устройствами.



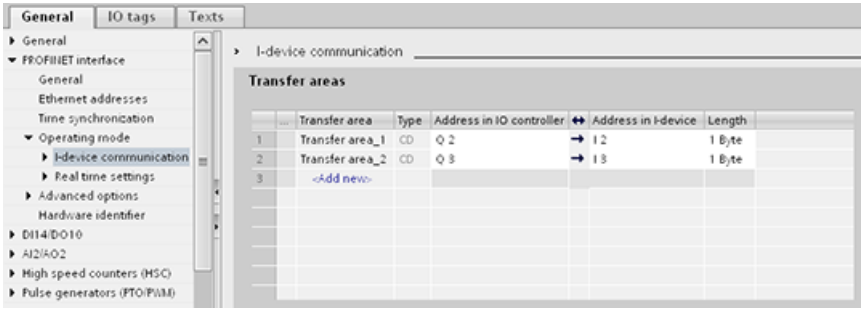
6. Флажок "Параметрирование PN интерфейса через IO-контроллер верхнего уровня" определяет, будут ли параметры интерфейсов назначаться самим I-устройством или IO-контроллером верхнего уровня.
   Если I-устройство работает в IO-системе нижнего уровня, то параметры PROFINET интерфейса I-устройства (напр., параметры портов) не могут назначаться IO-контроллером верхнего уровня.

7. Сконфигурировать области передачи. Области передачи можно найти в области навигации "Коммуникация интеллектуальных устройств":

   – Кликнуть мышью по первому полю столбца "Область передачи". STEP 7 присваивает стандартное имя, которое может быть изменено.

   – Выбрать тип коммуникационной связи: В настоящее время можно выбрать только CD или F-CD.

   – Адреса предустанавливаются автоматически. При необходимости можно исправить адреса и определить длину области передачи, которая должна быть передана последовательно.



8. Отдельная запись создается в области навигации для каждой области передачи. После выбора одной из этих записей, можно настраивать, исправлять или комментировать информацию об области передачи.

Примечание

При конфигурировании S7-1200 как I-устройства, макс. размер области передачи составляет 1024 входных или выходных байт. В зависимости от локального ввода-вывода и ограничений для адресного пространства на управляющем устройстве, могут возникнуть ограничивающие факторы.

Конфигурирование I-устройства с помощью файла GSD

При использовании I-устройства в другом проекте или в другой системе технических разработок, конфигурирование IO-контроллера верхнего уровня и I-устройства следует производить, как описано выше.

После конфигурирования областей передачи нажат кнопку "Экспорт" для создания нового файла GSD для I-устройства. Этот GSD-файл представляет сконфигурированное I-устройство в других проектах.

Кнопка "Экспорт" находится в области "Коммуникация интеллектуальных устройств" окна инспектора.

Конфигурация оборудования компилируется, и открывается диалоговое окно экспорта.

Назначить имя для прокси I-устройства, а также ввести описание в предусмотренных для этого полях. Нажать кнопку "Экспорт", чтобы завершить процесс.

После можно импортировать GSD-файл, например, в другой проект.

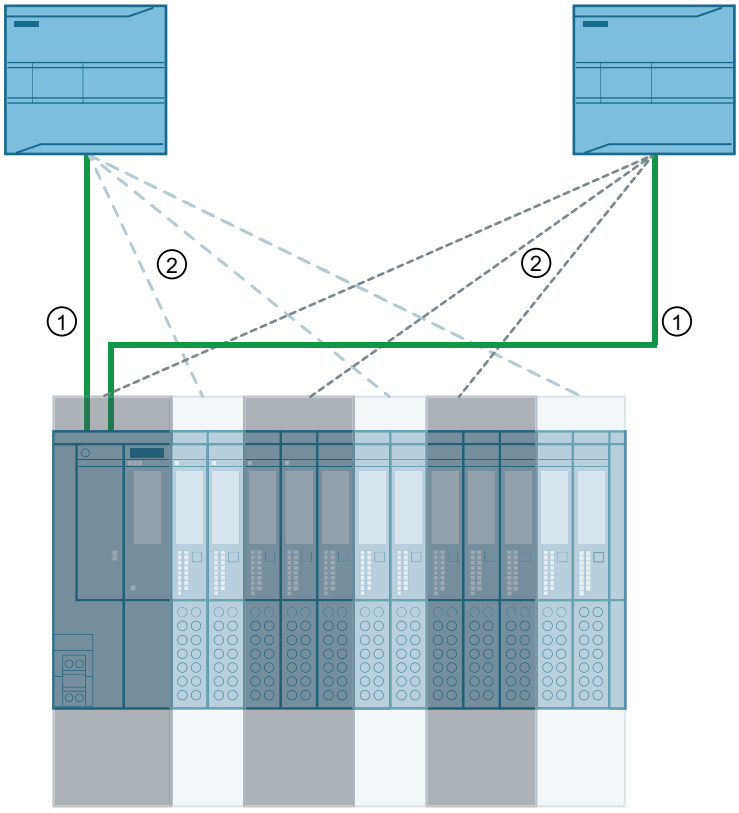
11.2.14 Устройства общего доступа

11.2.14.1 Функциональные возможности устройства общего доступа

В больших и широко рассредоточенных системах часто используется много IO-контроллеров.

Без функции "Устройство общего доступа" каждый модуль ввода-вывода IO-устройства назначен одному тому же IO-контроллеру. Если датчики, которые физически расположены близко друг к другу, должны предоставлять данные различным IO-контроллерам, то потребуется несколько IO-устройств.

Функция "Устройство общего доступа" позволяет распределить модули или субмодули IO-устройства между различными IO-контроллерами. Это расширяет возможности для гибкой автоматизации. Например, можно объединить модули ввода-вывода, находящиеся рядом друг с другом, в одно IO-устройство.



① PROFINET
② Логическое назначение

Принцип

Доступ к субмодулям устройства общего доступа в этом случае распределен по отдельным IO-контроллерам. Каждый субмодуль устройства общего доступа назначен только одному IO-контроллеру.

Необходимое условие (GSD конфигурация)

- STEP 7 V12 Service Pack 1 или выше
- S7-1200 CPU с версией прошивки от V4.1 как IO-контроллер
- IO-устройство поддерживает функцию устройства общего доступа, например, интерфейсный модуль IM 155-5 PN ST
- Файл GSD для конфигурации I-устройства установлен
- Сконфигурированный в качестве I-устройства S7-1200 CPU поддерживает функциональность устройства общего доступа. Необходимо экспортировать GSD-файл PROFINET для I-устройства из STEP 7 (от V5.5) и после импортировать его в STEP 7 (TIA Portal).

Конфигурирование доступа

IO-устройство должно присутствовать в нескольких проектах так, чтобы модули или субмодули IO-устройства могли быть назначены различным IO-контроллерам. Отдельный проект требуется для каждого IO-контроллера.

С помощью параметра "Устройства общего доступа" интерфейсного модуля определяются модули или субмодули, к которым IO-контроллер имеет доступ:

- Если локальный IO-контроллер обладает доступом к сконфигурированному модулю, то следует выбрать имя IO-контроллера из списка.
- Если IO-контроллер из другого проекта, а не локальный IO-контроллер должен обладать доступом к сконфигурированному модулю, выбрать элемент "---".

Конфигурация является согласованной относительно доступа, если каждый модуль или субмодуль назначен IO-контроллеру только в одном проекте.

Модуль или субмодуль назначены другому IO-контроллеру

Раздел ниже описывает следствия настройки "---" параметра "Устройства общего доступа" с точки зрения локального IO-контроллера.

В этом случае у локального IO-контроллера нет доступа к модулю, сконфигурированному таким образом. В частности это означает:

- Отсутствие обмена данными с модулем или субмодулем
- Отсутствие приема аварийных или диагностических сообщений, что означает отсутствие отображения состояния диагностики в онлайн просмотре
- Невозможность назначения параметров модуля или субмодуля

Настройка свойств реального времени

STEP 7 вычисляет коммуникационную нагрузку и соответствующее время обновления. Необходимо ввести число внешних IO-контроллеров для проекта, в котором PROFINET-интерфейс устройства общего доступа назначен IO-контроллеру, чтобы расчет был возможен с конфигурациями устройств общего доступа.

Максимальное возможное число IO-контроллеров для устройства общего доступа зависит от его типа. Это число сохранено в GSD-файле устройства общего доступа.

Можно установить очень короткий такт передачи (минимум 1 мс) с S7-1200 CPU в качестве IO-контроллера. Такт передачи может быть короче, чем наименьший так передачи, поддерживаемый устройством общего доступа. В этом случае устройство общего доступа управляется IO-контроллером с тактом передачи, который оно поддерживает (адаптация такта передачи).

Пример: CPU поддерживает такты передачи от 1 мс. Сконфигурированное IO-устройство поддерживает такты передачи от 1,25 мс, другое IO-устройство поддерживает такты передачи от 1 мс. В этом случае для CPU можно установить короткий такт передачи в 1 мс. CPU управляет "медленным" IO-устройством с тактом передачи 1,25 мс.

Правила для конфигурирования

- IO-контроллеры, которые используют устройства общего доступа, создаются в разных проектах. В каждом проекте нужно проследить, чтобы устройство общего доступа было сконфигурировано идентично на каждой станции. Только один единственный IO-контроллер может всегда иметь полный доступ к субмодулю. Противоречия в конфигурации приводят к ошибке устройства общего доступа.
- Адреса ввода-вывода модуля или субмодуля могут быть отредактированы только, если модуль или субмодуль назначены IO-контроллеру в том же проекте.
- У устройства общего доступа должны быть одни и те же IP-параметры и имя устройства в каждом проекте.
- Такт передачи должен быть идентичным для всех IO-контроллеров, которые могут обращаться к устройству общего доступа.
- Идентификатор S7-подсети, к которой подключено устройство общего доступа, должен быть идентичным во всех проектах.
- Следующие функции доступны только, если PROFINET-интерфейс устройства общего доступа назначен локальному IO-контроллеру:
  - Пуск по приоритету
  - Параметрирование свойств порта

Граничные условия

Следующие граничные условия имеют место, так как конфигурация с устройством общего доступа распределена по нескольким проектам:

- Адреса модулей или субмодулей, которые не назначены этому IO-контроллеру, отсутствуют в обзоре адресов каждого IO-контроллера, у которого есть доступ к устройству общего доступа.
- Не назначенные модули и субмодули не учитываются при расчете предельных значений для устройства общего доступа во время проверки непротиворечивости. Поэтому пользователь должен самостоятельно убедиться, что максимальное количество субмодулей или максимальный объем циклических IO-данных для устройства общего доступа не будут превышены. Для получения информации о максимальных значениях следует обратиться к документации на используемые устройства.
- Ошибки конфигурации, такие как назначение модуля или субмодуля нескольким IO-контроллерам, не обнаруживаются в STEP 7.
- У CPU, в которые были загружены конфигурации с устройством общего доступа, нет информации о том, является ли данные IO-устройство устройством общего доступа. Модули или субмодули, которые назначены другим IO-контроллерам и, таким образом, другим CPU, отсутствуют в загруженной конфигурации. Следовательно, такие модули или субмодули не отображаются ни в веб-сервере CPU, ни на дисплее CPU.

11.2.14.2 Пример: Конфигурирование устройства общего доступа (GSD конфигурация)

Этот пример показывает, как сконфигурировать распределенную систему ввода-вывода в качестве устройства общего доступа с помощью STEP 7 V13 SP1 или выше.

Возможна "распределенная" конфигурация с различными инженерными инструментами для различных семейств IO-контроллеров. Процедура, описанная ниже, основывается на STEP 7 V13 SP1 и ограничивается конфигурацией с двумя IO-контроллерами серий S7-1200, которые вместе используют одно устройство общего доступа.

В примере создаются два проекта с одним IO-контроллером в каждом:

- Controller1
- Controller2

В обоих проектах должно быть создано устройство общего доступа, даже в том случае, когда физически - это одно и то же IO-устройство.

Необходимые условия

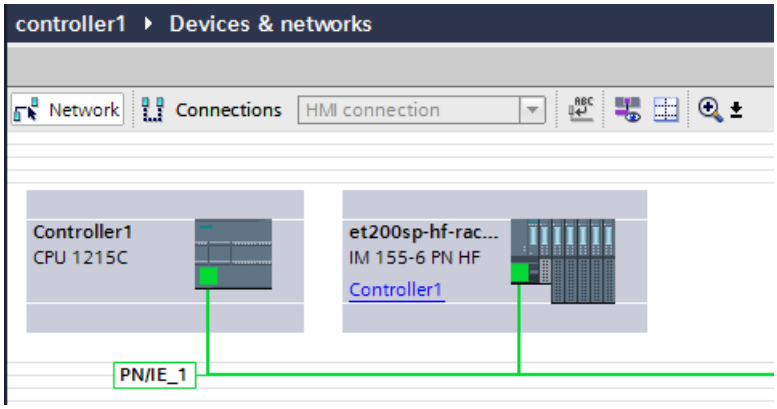
- STEP 7 от версии V13 SP1
- IO-устройство поддерживает функцию устройства общего доступа (например, ET 200SP IM 155-6 PN HF V3.1)
- Файл GSD для конфигурации IO-устройства как устройства общего доступа установлен.

Порядок действий: Создание проекта 1

Для создания первого проекта с устройством общего доступа, выполнить следующие шаги:

1. Запустить STEP 7.
2. Создать новый проект с именем "Controller1".
3. Вставить CPU 1215C из каталога оборудования в просмотре сетевых соединений. Назвать его "Controller1".
4. Вставить IO-устройство с функциональностью "Устройство общего доступа" (напр., ET 200SP) из каталога оборудования (каталог оборудования: Другие полевые устройства > PROFINET IO > Ввод/вывод).

5. Назначить IO-контроллер "Controller1" IO-устройству.

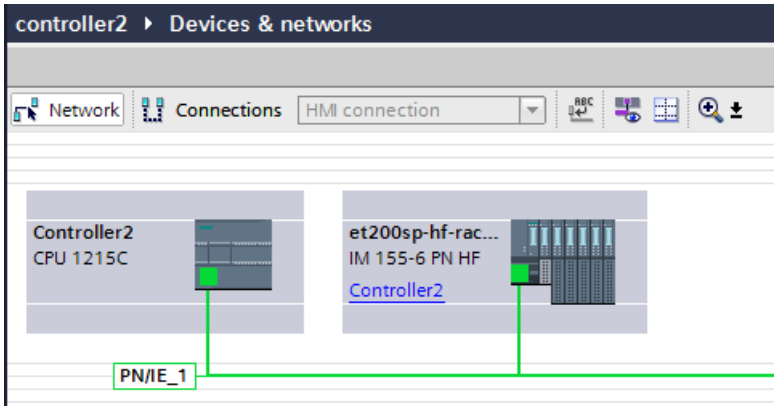


6. Дважды кликнуть по IO-устройству и вставить все требуемые модули и субмодули из каталога оборудования в таблицу обзора устройств.
7. Назначить параметры модулей.
8. Сохранить проект.

Порядок действий: Создание проекта 2

Для создания второго проекта с устройством общего доступа, выполнить следующие шаги:

1. Заново запустить STEP 7.
   Откроется новый экземпляр STEP 7.
2. Создать в новом экземпляре новый проект с именем "Controller2".
3. Вставить CPU 1215C в просмотре сетевых соединений. Назвать его "Controller2".
4. Скопировать IO-устройства из проекта "Controller1" и вставить его в просмотре сетевых соединений проекта "Controller2".
5. Назначить IO-контроллер "Controller2" IO-устройству.



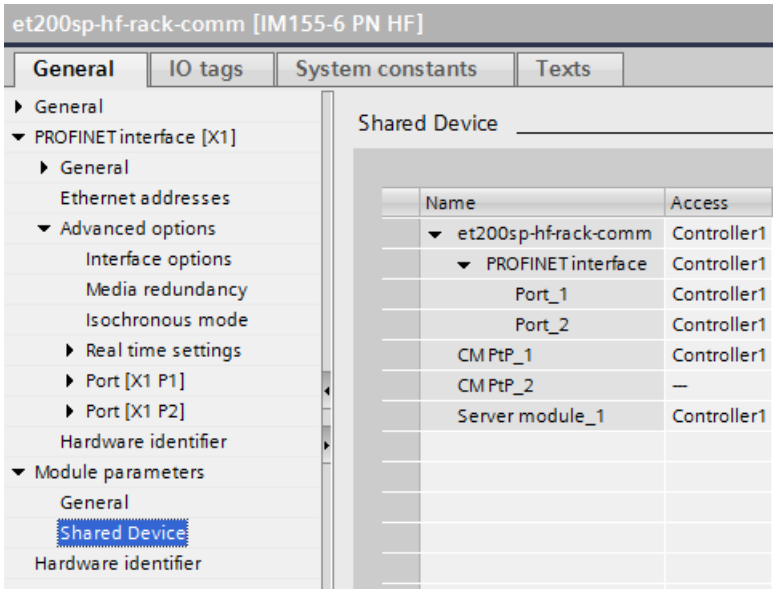
6. Сохранить проект.

В обоих проектах теперь есть тождественно структурированное IO-устройство, которое должно быть сконфигурировано на следующем шаге для других типов доступа IO-контроллеров.

Порядок действий: Конфигурирование обращений к устройству общего доступа

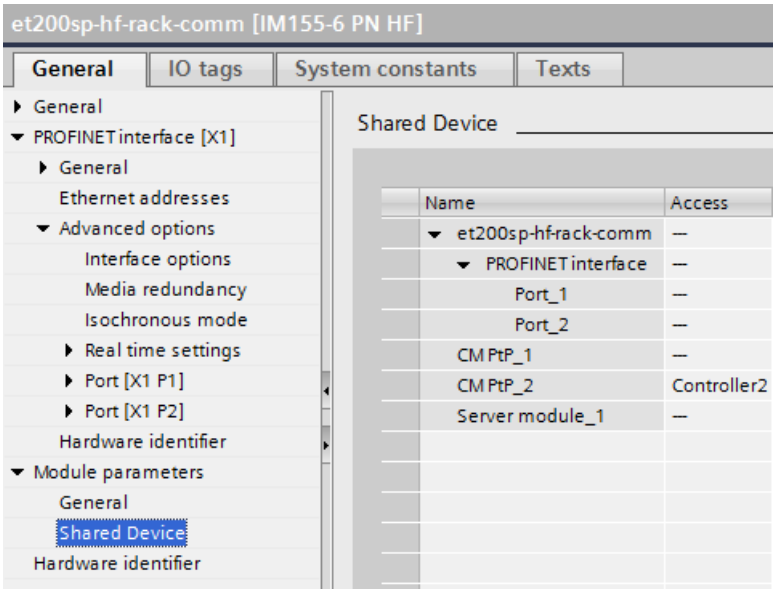
Модули и субмодули, которые вставляются в устройство общего доступа, автоматически назначаются локальному CPU. Чтобы изменить назначение, выполнить следующие шаги:

1. Выбрать интерфейсный модуль в просмотре сетевых соединений или просмотре устройств для проекта "Controller1"
2. Выбрать область "Устройство общего доступа" в окне инспектора.
   Таблица показывает, у какого CPU есть доступ к модулям и субмодулям для всех сконфигурированных модулей. По умолчанию у локального CPU есть доступ ко всем модулям и субмодулям.
3. Сохранить настройку "Controller1" для всех модулей и субмодулей, которые должны остаться в диапазоне адресов локального CPU.
   Выбрать настройку "---" для всех модулей и субмодулей, которые должны находиться в диапазоне адресов CPU из проекта "Controller2". Это означает, что IO-контроллер вне проекта получает доступ к модулю или субмодулю.



4. Выбрать интерфейсный модуль в просмотре сетевых соединений или просмотре устройств для проекта "Controller2"
5. Выбрать область "Устройство общего доступа" в окне инспектора.
   Таблица показывает, у какого CPU есть доступ к модулям и субмодулям для всех сконфигурированных модулей.

6. Выбрать настройку "---" для всех модулей и субмодулей, которые должны находиться в диапазоне адресов CPU из проекта "Controller1".



7. В конце следует проверить, "согласованны" ли настройки доступа для каждого отдельного модуля или субмодуля в обоих проектах. Это означает, что, если у локального CPU есть доступ в одном проекте, то опция "---" должна быть установлена в другом проекте и наоборот.
   Примечание: Опция "---" для PROFINET-интерфейса и, следовательно, для портов, переводит связанные параметры в состояние только для чтения, блокируя их редактирование. Параметры PROFINET-интерфейса и параметры портов могут быть отредактированы только в проекте, в котором PROFINET-интерфейс назначен локальному CPU. Порты могут быть соединены в обоих проектах друг с другом независимо от этого условия.

8. Проверить, установлены ли одни и те же параметры IP-адреса и одно имя устройства для устройства общего доступа во всех проектах.
   Проверить, установлен ли один тот же идентификатор S7-подсети во всех проектах для подсети, к которой подключено устройство общего доступа (свойства подсети, область "Общее" в окне инспектора).

Примечание

При внесении изменений в устройство общего доступа: Выполнить одни и те же изменения в каждом проекте на устройстве общего доступа. Проследить, что только у одного IO-контроллера есть доступ к модулю или субмодулю.

Порядок действий: Адаптация параметров реального времени

Чтобы гарантировать, что все IO-контроллеры и устройства общего доступа работают с подходящим тактом передачи и, что время обновления правильно рассчитывается на основе коммуникационной нагрузки, необходимо проверить и согласовать следующие настройки:

1. Выбрать проект, в котором IO-контроллеры обладают доступом к PROFINET-интерфейсу и портам устройства общего доступа.
2. Выбрать в просмотре сетевых соединений интерфейсный модуль устройства общего доступа.
3. Перейти в окне инспектора к области "Интерфейс PROFINET > Расширенные опции > Настройки реального времени > IO-цикл".
4. В области "Устройство общего доступа" задать количество внешних для проекта IO-контроллеров. Максимальное количество зависит от IO-устройства (указано в GSD-файле).
5. Необходимо установить идентичный такт передачи для каждого IO-контроллера, у которого есть доступ к модулям и субмодулям устройства общего доступа:

- Если для конфигурирования IO-контроллера используется STEP 7 (TIA Portal):
  - Открыть соответствующий проект.
  - Выбрать PROFINET-интерфейс IO-контроллера.
  - Выбрать область "Расширенные опции > Настройки реального времени > IO-коммуникация" в окне инспектора и задать такт передачи.
- Если для конфигурирования IO-контроллера используется другой инженерный инструмент:
  - Выбрать интерфейс PROFINET устройства общего доступа в STEP 7 (TIA Portal) и считать такт передачи устройства общего доступа (область "Расширенные опции > Настройки реального времени").
  - Ввести считанный такт передачи в инженерный инструмент.

Примечание

Если конфигурируются все IO-контроллеры, которые имеют доступ к устройству общего доступа, в STEP 7 (TIA Portal), то можно установить на IO-контроллере более короткие такты передачи, чем поддерживает устройство общего доступа (адаптация такта передачи).

Компиляция и загрузка

Необходимо скомпилировать конфигурации для различных IO-контроллеров и загрузить их в CPU одну за другой.

Из-за распределенной конфигурации с отдельными проектами, STEP 7 не выводит ошибки из-за несовместимости в случае неправильного параметрирования доступа. Здесь некоторые примеры неправильного параметрирования доступа:

- Несколько IO-контроллеров обладают доступом к одному и тому же модулю
- Параметры IP-адреса или такты передачи не идентичны

Эти ошибки отображаются только при работе контроллера и сигнализируются как ошибки конфигурации.

11.2.14.3 Пример: Конфигурирование I-устройства как устройства общего доступа

Этот пример показывает, как сконфигурировать S7-1200 в качестве I-устройства с помощью STEP 7 V13 SP1 и после использовать его в двух проектах в качестве устройства общего доступа

Возможна "распределенная" конфигурация с различными инженерными инструментами для различных семейств IO-контроллеров. Процедура, описанная ниже, основывается на STEP 7 V13 SP1 и ограничивается конфигурацией с двумя IO-контроллерами серий S7-1200, которые совместно используют области передачи I-устройства как устройства общего доступа. Само I-устройство - это CPU 1215C.

В примере создаются три проекта с одним IO-контроллером в каждом:

- S7-1200-I-Device
- Controller1
- Controller2

С помощью проекта S7-1200-I-Device конфигурируется I-устройство. Используется PROFINET GSD вариант S7-1200-I-Device в проектах Controller1 и Controller2, чтобы назначить области передачи в соответствующем IO-контроллере верхнего уровня.

Концепция I-устройства общего доступа

Для концепции I-устройства общего доступа потребуется как минимум три отдельных проекта:

- Проект I-устройства: I-устройство конфигурируется и программируется для выполнения определенной задачи автоматизации. Области передачи определяются как интерфейс ввода-вывода для контроллеров верхнего уровня и эти области передачи назначаются различным IO-контроллерам. Для соединения с IO-контроллерами верхнего уровня предоставляется PROFINET GSD файл и области передачи используются, чтобы получить доступ к I-устройству.
- Контроллеры, которые совместно используют I-устройство (два проекта): I-устройство используется в качестве PROFINET GSD варианта во время конфигурирования PROFINET IO системы и, в ходе этого процесса, указываются адреса ввода-вывода, по которым IO-контроллеры получают доступ к областям передачи.

Примечание

При конфигурировании S7-1200 как I-устройства, макс. размер области передачи составляет 1024 входных или выходных байт. В зависимости от локального ввода-вывода и ограничений для адресного пространства на управляющем устройстве, могут возникнуть ограничивающие факторы.

I-устройство

Следующие параметры назначаются S7-1200 CPU в качестве I-устройства:

- Централизованная и распределенная периферия
- Требуемые области передачи
- Количество IO-контроллеров, имеющих доступ к этому I-устройству (всегда больше, чем 1 для устройства общего доступа)

Примечание

I-устройство конфигурируется без IO-контроллера верхнего уровня. Результатом является то, что можно использовать только локальные адреса ввода-вывода области передачи (= "Адрес в I-устройстве"), чтобы создать пользовательскую программу для редактирования адресов из области передачи. I-устройство загружается в S7-1200 CPU полностью сконфигурированным, за исключением соединения с IO-контроллером верхнего уровня.

PROFINET GSD файл экспортируется из конфигурации I-устройства.

Контроллеры, которые совместно используют I-устройство

Необходимо загрузить PROFINET GSD файл, созданный из конфигурации I-устройства, во все инженерные системы, которые используются при конфигурировании PROFINET IO системы с этим I-устройством общего доступа. Если для конфигурирования всех случаев использования этого I-устройства используется STEP 7 V13 SP1, то достаточно установить GSD-файл в STEP 7.

I-устройство конфигурируется как GSD вариант на PROFINET IO системе в задействованных проектах. В STEP 7 V13 SP1 после установки это I-устройство находится в разделе "Другие полевые устройства" > PROFINET IO > PLC и CP" .

В каждом из задействованных проектов области передачи назначаются исключительно IO-контроллерам верхнего уровня (настройка по умолчанию: все). Для других областей передачи устанавливается "---" (не назначено). Вследствие этого локальный IO-контроллер не может обращаться к этой области передачи, и область передачи может быть назначена другому IO-контроллеру в другом проекте.

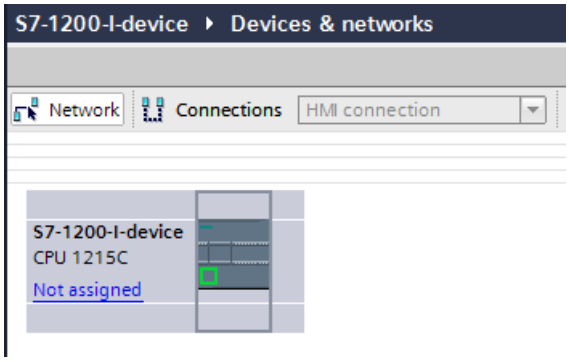
Необходимые условия

- STEP 7 от версии V13 SP1
- IO-устройство поддерживает функциональность устройства общего доступа (например, ET 200SP IM 155-6 PN HF V3.1)
- Файл GSD для конфигурации IO-устройства как устройства общего доступа установлен.

Порядок действий: Создание проекта S7-1200-I-Device

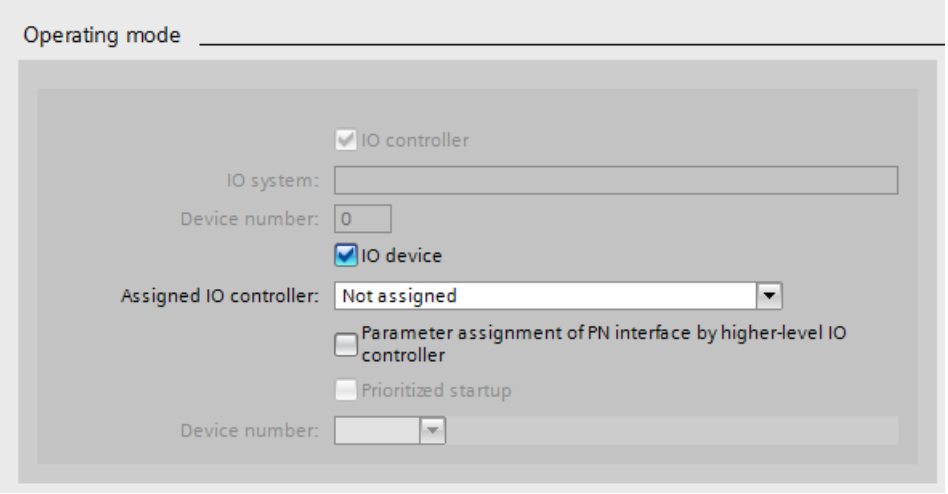
Для создания с I-устройством общего доступа, выполнить следующие шаги:

1. Запустить STEP 7.
2. Создать новый проект с именем "S7-1200-I-Device".
3. Вставить CPU 1215C из каталога оборудования в просмотре сетевых соединений. Присвоить имя "S7-1200-I-Device".

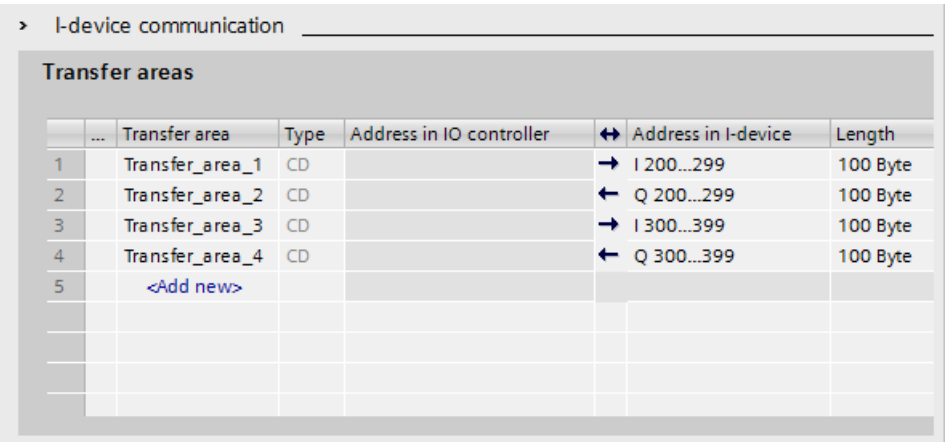


4. Дважды кликнуть по IO-устройству и сконфигурировать все требуемые модули и субмодули.

5. Назначить параметры модулей. В частности, необходимо сконфигурировать следующие параметры для CPU в области PROFINET интерфейса [X1]:
   – Активировать опцию "IO-устройство" в области "Режим работы".

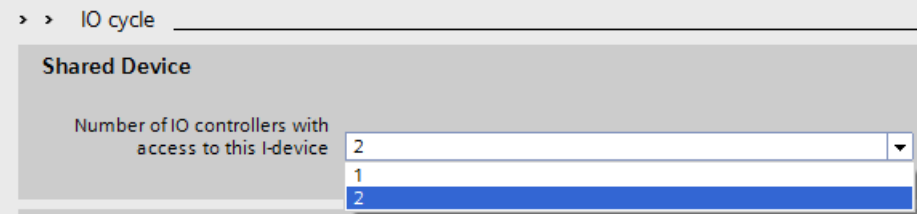


   – Сконфигурировать области передачи в области "Режим работы" > "Конфигурация I-устройства". Столбец "Адрес в IO-контроллере" остается пустым, потому что IO-контроллер не назначен.



   Примечание: Чтобы изменить область ввода на область вывода, и наоборот, следует перейти к соответствующей области передачи.

   – Выбрать количество IO-контроллеров (минимум два), которые при работе обращаются к I-устройству общего доступа (область "Режим работы > Настройки реального времени", область "Устройство общего доступа").



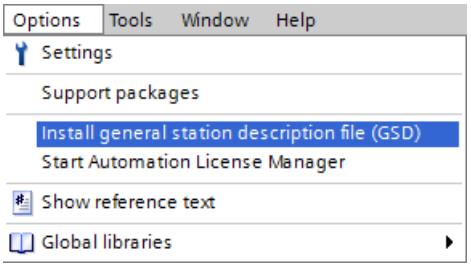
6. Сохранить проект.
7. Нажать кнопку "Экспорт" (область "Режим работы > Конфигурация I-устройства", раздел "Экспорт файла описания основных данных устройства (GSD)"). Если не изменять имя в диалоговом окне экспорта, то файл GSD получает имя в назначенном формате (например, "GSDML-V2.31-#Siemens-PreConf_S7-1200-I-Device-20130925-123456").



Порядок действий: Создание проекта Controller1

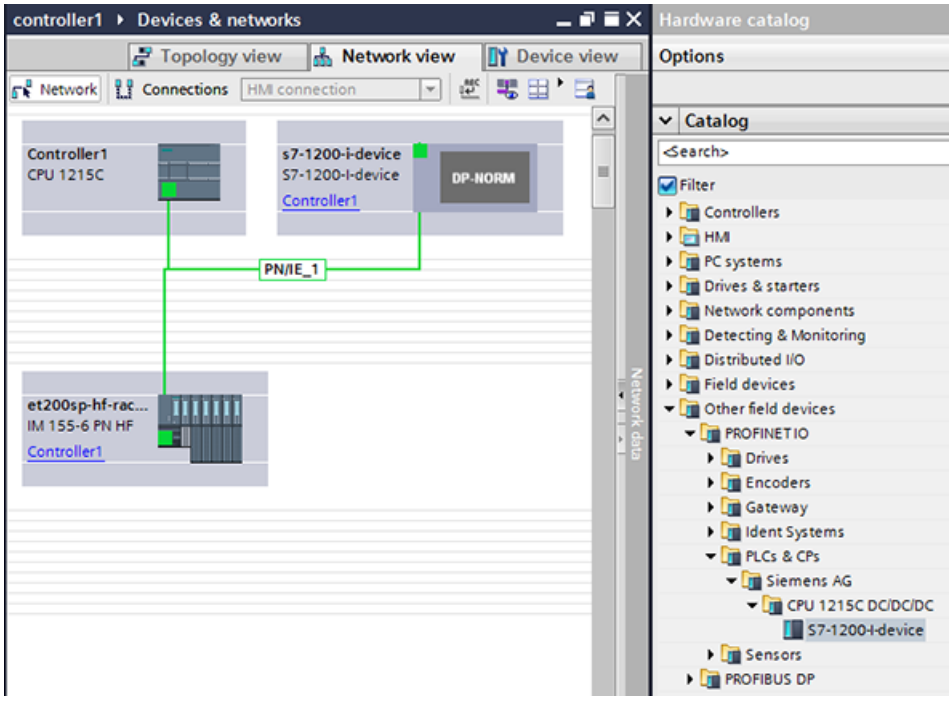
Для создания первого проекта с I-устройством общего доступа, выполнить следующие шаги:

1. Запустить STEP 7.
2. Установить экспортированный из I-Device-CPU файл PROFINET GSD (S7-1200-I-Device).

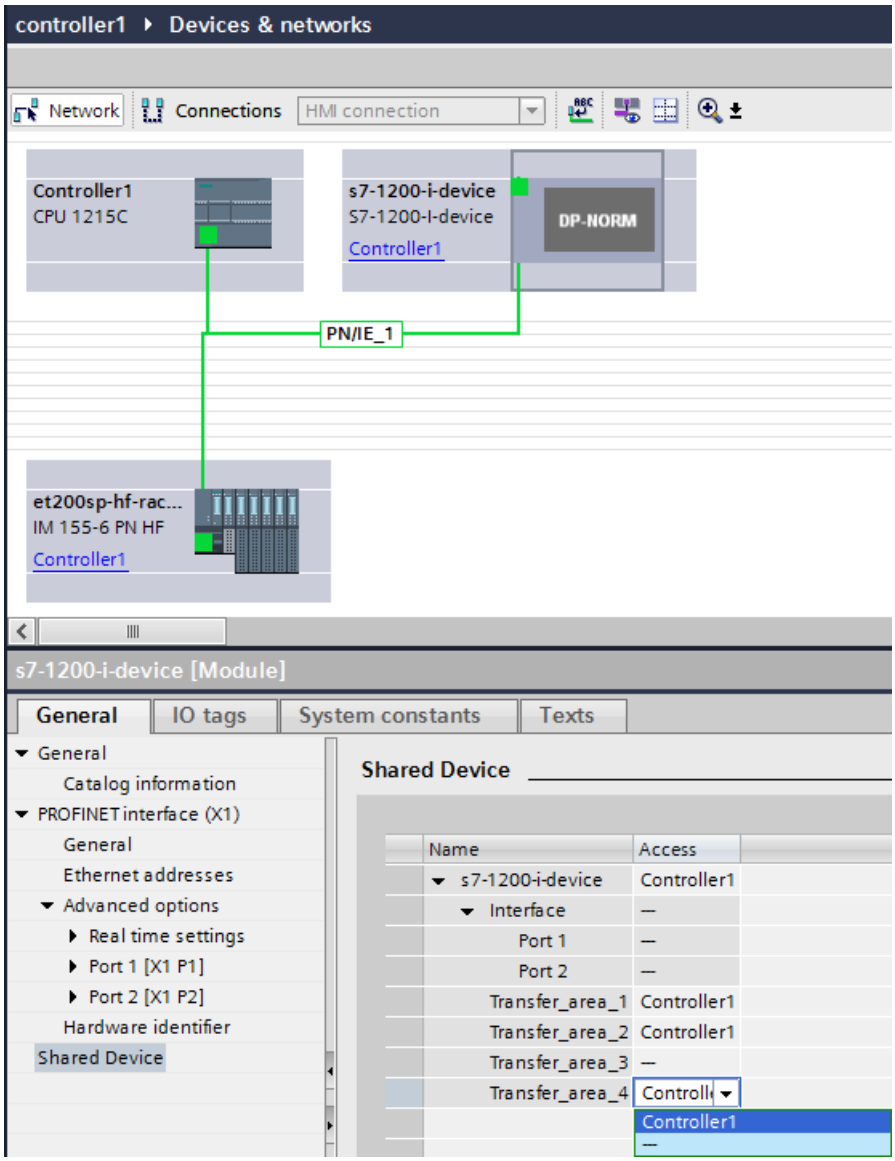


3. Создать новый проект с именем "Controller1".
4. Вставить CPU 1215C в просмотре сетевых соединений. У CPU должно быть имя "Controller1".
5. Вставить I-устройство из каталога оборудования (каталог оборудования: Другие полевые устройства > PROFINET IO > PLC и CP).

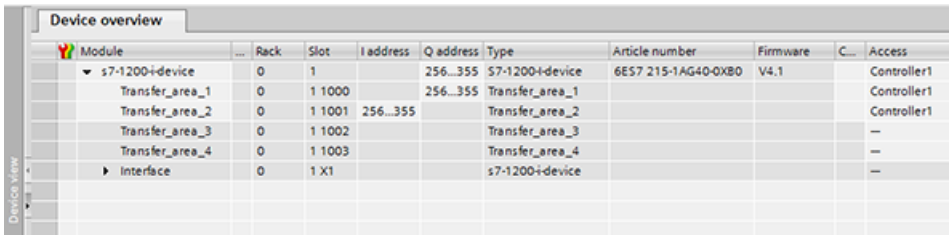
6. Назначить IO-контроллер "Controller1" I-устройству.



7. Выбрать область "Устройство общего доступа" в свойствах I-устройства:
   - В таблице все области передачи и интерфейс PROFINET назначены локальному IO-контроллеру (Controller1).
   - Определить области передачи, к которым Controller1-CPU **не** сможет обращаться. Выбрать для этих областей элемент "---". Эти области передачи предоставляются устройству Controller2.



8. Адреса из просмотра устройств IO-контроллера можно изменять в обзоре устройства. Чтобы открыть обзор устройства, дважды кликнуть по I-устройству.



9. Сохранить проект.

Порядок действий - Создание проекта Controller2

Для создания второго проекта с устройством общего доступа, выполнить следующие шаги:

1. Заново запустить STEP 7.
   Откроется новый экземпляр STEP 7.
2. Создать в новом экземпляре новый проект с именем "Controller2".
3. Вставить CPU 1215C в просмотре сетевых соединений. Присвоить имя "Controller2".
4. Вставить I-устройство из каталога оборудования (каталог оборудования: Другие полевые устройства > PROFINET IO > PLC и CP).
5. Назначить IO-контроллер "Controller2" I-устройству.
6. Настроить доступ к областям передачи в проекте Controller1. Убедиться, что двойные назначения отсутствуют.
7. Настроить параметры подсети и PROFINET-интерфейса. Т.к. I-устройство общего доступа использует одной и то же устройство в разных проектах, эти данные должны совпадать.
8. Сохранить проект.

Теперь в обоих проектах есть идентично сконфигурированное I-устройство общего доступа. Но доступ IO-контроллера и параметры PROFINET-интерфейса все же должны быть проверены в разных проектах во время следующего шага.

Итоговая информация - Параметрирование обращений к устройству общего доступа

Области передачи автоматически назначаются локальному IO-контроллеру. Чтобы изменить назначение, выполнить следующие шаги:

1. Кликнуть мышью в просмотре сетевых соединений проекта "Controller1" по устройству "S7-1200-I-Device" и выбрать область "Устройство общего доступа".
2. Таблица показывает, у какого CPU есть доступ к отдельным сконфигурированным областям передачи. По умолчанию у локального CPU есть доступ ко всем модулям и субмодулям.

3. Сохранить настройку "Controller1" для всех областей передачи, которые должны остаться в диапазоне адресов локального CPU.
   Выбрать настройку "---" для всех областей передачи, которые должны находиться в диапазоне адресов CPU "Controller2" из проекта "Controller2". Это означает, что IO-контроллер вне проекта получает доступ к области передачи.
4. Выполнить ту же процедуру для оставшихся проектов.
5. В конце следует проверить, "согласованны" ли настройки доступа для каждого отдельного модуля или субмодуля в обоих проектах. Это означает, что, если у локального CPU есть доступ в одном проекте, то опция "---" должна быть установлена в другом проекте и наоборот.
   Примечание: Опция "---" для PROFINET-интерфейса и, следовательно, для портов, переводит связанные параметры в состояние только для чтения, блокируя их редактирование. Параметры PROFINET-интерфейса и параметры портов могут быть отредактированы только в проекте, в котором PROFINET-интерфейс назначен локальному CPU. Порты могут быть соединены в обоих проектах друг с другом независимо от этого условия.
6. Проверить, установлены ли одни и те же параметры IP-адреса и одно имя устройства для устройства общего доступа во всех проектах.
   Проверить, установлен ли один тот же идентификатор S7-подсети во всех проектах для подсети, к которой подключено устройство общего доступа (свойства подсети, область "Общее" в окне инспектора).

Примечание

После внесения изменений в I-устройство (например, после изменения количества или размера областей передачи), следует повторно экспортировать I-устройство как GSD-файл. Переустановить GSD-файл в каждом проекте, который использует I-устройство в качестве устройства общего доступа. Проследить, что только у одного IO-контроллера есть доступ к области передачи.

Примечание

Если S7-1200 используется как I-устройство общего доступа и как контроллер, то необходимо увеличить время обновления PROFINET I-устройства и PROFINET IO для компенсации влияния на коммуникационную способность. Система является чрезвычайно стабильной и работает правильно, если для отдельного PROFINET I-устройства и для отдельного PROFINET IO выбирается время обновления в 2 мс.

Параметры для "IO цикла" определяются в конфигурационном диалоге "Свойства" PROFINET IO устройства или ввода-вывода. Дополнительную информацию см. в "Конфигурирование времени цикла IO" (Страница 942).

Порядок действий - Адаптация параметров реального времени

Чтобы гарантировать, что все IO-контроллеры и устройства общего доступа работают с подходящим тактом передачи и, что время обновления правильно рассчитывается на основе коммуникационной нагрузки, необходимо проверить и согласовать следующие настройки:

1. Необходимо установить идентичный такт передачи для каждого IO-контроллера, у которого есть доступ к модулям и субмодулям устройства общего доступа:

- Если для конфигурирования IO-контроллера используется STEP 7 (TIA Portal), то действовать следующим образом:
  - Открыть соответствующий проект.
  - Выбрать PROFINET-интерфейс IO-контроллера.
  - Выбрать область "Расширенные опции > Настройки реального времени > IO-коммуникация" в окне инспектора и задать такт передачи.
- Если для конфигурирования IO-контроллера используется другой инженерный инструмент, то действовать следующим образом:
  - Выбрать интерфейс PROFINET устройства общего доступа в STEP 7 (TIA Portal) и считать такт передачи устройства общего доступа (область "Расширенные опции > Настройки реального времени").
  - Ввести считанный такт передачи в инженерный инструмент.

Примечание

Если конфигурируются **все** IO-контроллеры, которые имеют доступ к I-устройству общего доступа, в STEP 7 (TIA Portal), то можно установить на IO-контроллере более короткие такты передачи, чем поддерживает устройство общего доступа (адаптация такта передачи).

Компиляция и загрузка

Необходимо скомпилировать конфигурации для различных IO-контроллеров и загрузить их в CPU одну за другой.

Из-за распределенной конфигурации с отдельными проектами, STEP 7 не выводит ошибки из-за несовместимости в случае неправильного параметрирования доступа. Здесь некоторые примеры неправильного параметрирования доступа:

- Несколько IO-контроллеров обладают доступом к одному и тому же модулю.
- Параметры IP-адреса или такты передачи не идентичны.

Эти ошибки отображаются только при работе контроллера и сигнализируются как ошибки конфигурации.

11.2.15 Протокол резервирования среды (MRP)

Следующие S7-1200 CPU версии V4.2.x и выше поддерживают режим MRP-клиента, но не могут использоваться в роли MRP-менеджера:

- CPU 1215C
- CPU 1217C
- CPU 1215FC

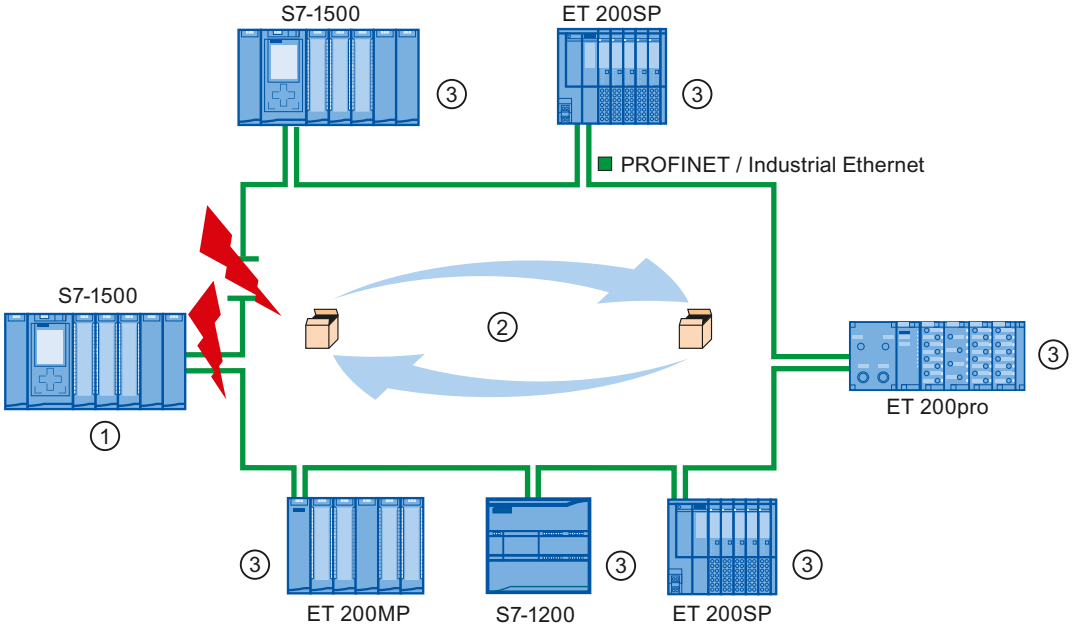
У трех S7-1200 CPU есть все два порта PN, необходимые для поддержки MRP протокола и параметров конфигурации для инициализации клиентского режима MRP.

11.2.15.1 Резервирование среды в кольцевых топологиях

Для увеличения доступности сети Industrial Ethernet с оптической или электрической линейной шинной топологией, можно преобразовать линейную шинную топологию в кольцевую топологию, соединив концы вместе.

Устройствами в кольцевой топологии могут быть IO-устройства, IO-контроллеры, внешние коммутаторы и/или встроенные коммутаторы коммуникационных модулей.

Для создания кольцевой топологии с резервированием среды необходимо соединить оба свободных конца линейной шинной топологии в одном устройстве. Для замыкания линейно шинной топологии и формирования кольца потребуется два порта (кольцевые порты) устройства в кольце. Одно из устройств в полученном кольце берет на себя роль MRP-менеджера. Все другие устройства в кольце являются MRP-клентами.



① MRP-менеджер
② Тестовые телеграммы
③ MRP-клиенты

Кольцевые порты устройства устанавливают соединение с двумя соседними устройствами в кольцевой топологии. Кольцевые порты выбираются и устанавливаются (при необходимости и предустанавливаются) в конфигурации соответствующего устройства).

Принцип работы резервирования среды в кольцевой топологии

Если кольцо в любом месте прерывается, то линии передачи данных между отдельными устройствами автоматически переконфигурируются. После переконфигурирования устройства снова доступны.

В MRP-менеджере один из двух кольцевых портов в штатном сетевом режиме заблокирован для обычной коммуникации, поэтому датаграммы через него не передаются. С точки зрения передачи данных, кольцевая топология является линейной шинной топологией. MRP-менеджер контролирует кольцо на предмет прерываний. Для этого он передает тестовые телеграммы (тестограммы) с кольцевого порта 1 и кольцевого порта 2. Тестовые телеграммы движутся в обоих направлениях по кольцу до прибытия на другой кольцевой порт MRP-менеджера.

Прерывание кольца возможно при потере соединения между двумя устройствами или отказе устройства в кольце.

Если тестограммы от MRP-менеджера при прерывании кольца больше не приходят на другой кольцевой порт, то MRP-менеджер соединяет оба своих кольцевых порта. Такой замещающий путь восстанавливает соединение между всеми остальными устройствами в форме линейной шинной топологии.

Время между разрывом кольца и восстановлением работоспособной линейной топологии называется временем переконфигурирования.

Сразу после устранения прерывания, исходные пути передачи восстанавливаются, два кольцевых порта MRP-менеджера разъединяются, и MRP-клиенты получают информацию об изменениях. После MRP-клиенты снова используют исходные пути к другим устройствам.

Метод резервирования среды

Стандартным методом резервирования среды в SIMATIC является протокол резервирования среды (MRP) со стандартным временем переконфигурирования в 200 мс. Кольцо может включать в себя до 50 устройств.

11.2.15.2 Использование протокола резервирования среды (MRP)

Процесс MRP использует протокол резервирования среды (MRP), специфицированный в IEC 61158 Typ 10 "PROFINET".

Необходимые условия

Для правильной работы с MRP должны быть выполнены следующие условия:

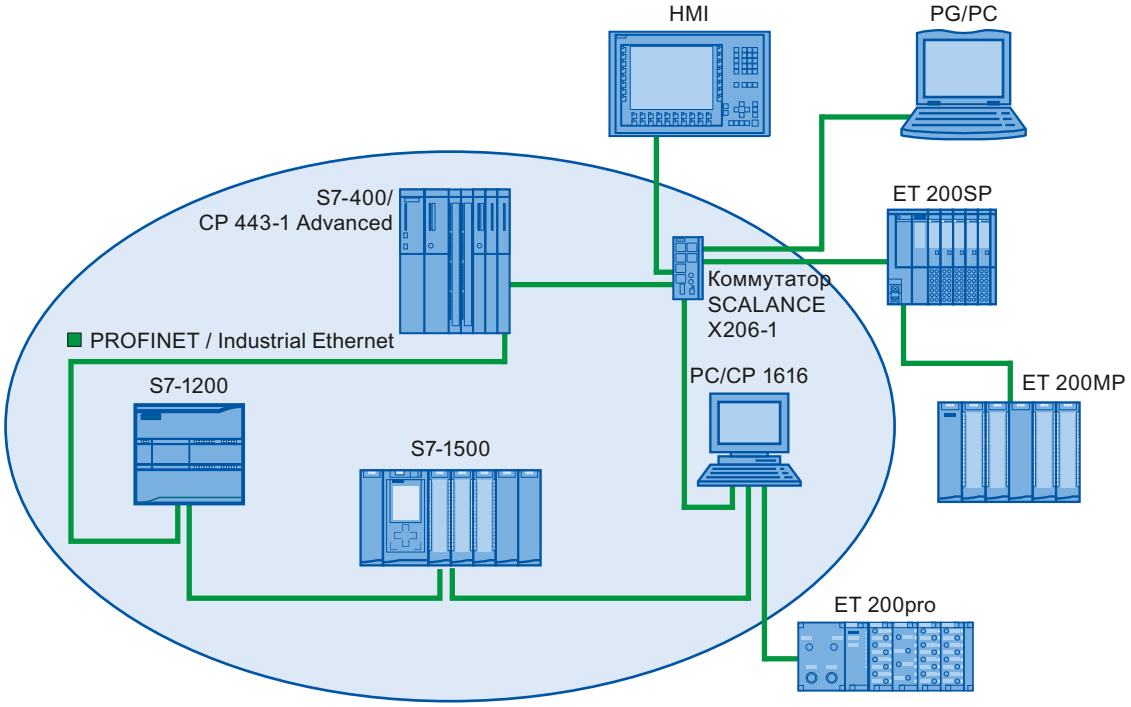
- Кольцо, в котором будет использоваться MRP, должно состоять только из устройств, которые поддерживают эту функцию.
- Для всех устройств кольца должен быть активирован протокол "MRP".

- Все устройства должны быть соединены друг с другом через свои кольцевые порты.
- Необходимо наличие как минимум одного MRP-менеджера (роль "Менеджер (автом.)").
- Кольцо может состоять максимум из 50 устройств. В ином случае время переконфигурирования может составить от 200 мс и выше.
- Все партнерские порты в кольце должны иметь идентичные настройки.

Топология

Следующее изображение показывает возможную топологию для устройств в кольце с MRP. Устройства внутри затемненного овала находятся в домене резервирования.

Это пример для кольцевой топологии с MRP:



Следующие правила действуют для кольцевой топологии с резервированием среды и MRP:

- Все устройства в кольце относятся к одному домену резервирования.
- Одно из устройств в кольце берет на себя роль MRP-менеджера.
- Все другие устройства в кольце являются MRP-клентами.

Устройства, не поддерживающие MRP, могут быть подключены к сети через порты, не сконфигурированные в качестве кольцевых портов. Такая возможность есть только у устройств более чем с двумя портами (напр., коммутатор SCALANCE X или PC с CP1616).

Граничные условия

Существуют две следующие коммуникационные возможности:

- MRP поддерживает режим MRP и RT: RT.

  Примечание

  RT-коммуникация прерывается (отказ станции), если временем переконфигурирования кольца превышает выбранное время контроля IO-устройства. Для IO-устройства должно быть выбрано время контроля больше чем 200 мс. Более подробная информация содержится в следующем разделе "Время контроля".

- MRP и TCP/IP (TSEND, HTTP, ...): TCP/IP-коммуникация с MRP возможна, так как потерянные пакеты данных при необходимости передаются заново.
- MRP и пуск по приоритету:
  - Если MRP конфигурируется в кольце, то для участвующих устройств нельзя использовать функцию "Пуск по приоритету" в приложениях PROFINET.
  - Для использования функции "Пуск по приоритету" необходимо выключить MRP в конфигурации (устройство не может быть частью кольца).
- MRP на устройствахPROFINET с числом портов больше двух: Если используется устройство PROFINET с более чем двумя портами в одном кольце, то необходимо установить предел синхронизации на портах, не относящихся к кольцу. Установка предела синхронизации определяет границу для доменов синхронизации. Нельзя пересылать телеграммы для синхронизации устройств в домене синхронизации.

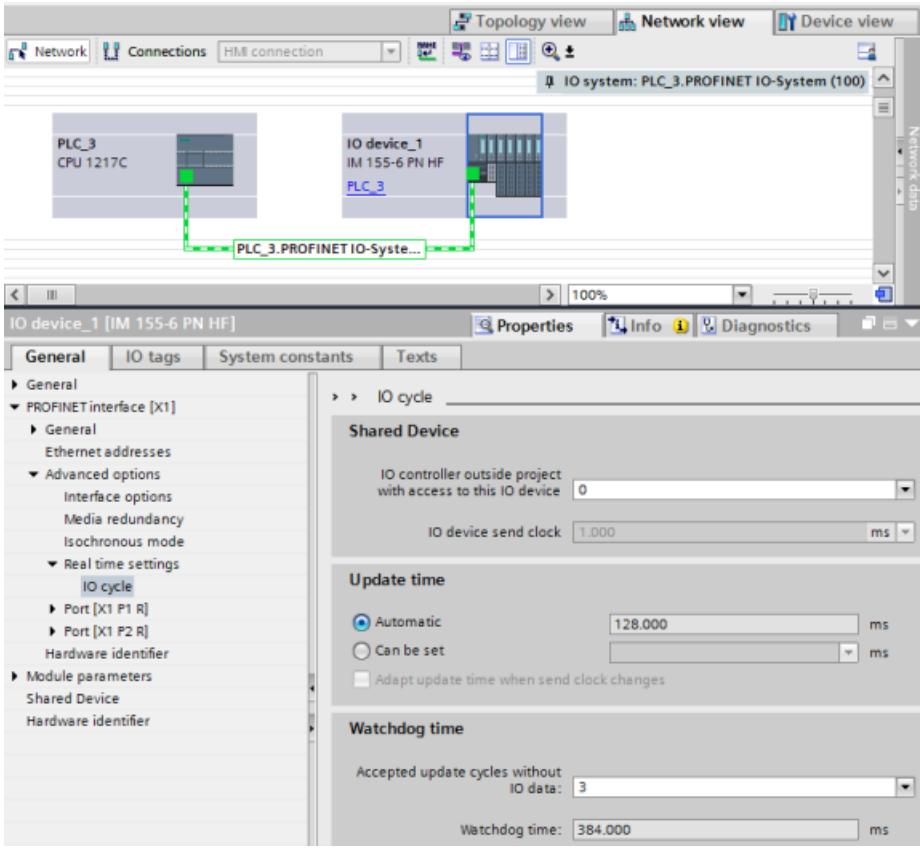
Время контроля

Время контроля - это допустимый интервал времени, в течение которого IO-контроллер или IO-устройство могут не получать IO-данные. Если IO-устройство не получит от IO-контроллера данными в течение времени контроля, то устройство обнаруживает пропущенные телеграммы и использует замещающие значения. Это сигнализируется на IO-контроллере как отказ станции.

Время контроля для PROFINET IO-устройств может конфигурироваться. Время контроля должно вводиться не напрямую, а как "Допустимое число циклов обновления при отсутствии IO-данных". Полученное время контроля после вычисляется автоматически из числа циклов обновления.

Для назначения времени контроля выполнить следующие действия:

1. Выбрать PROFINET-интерфейс IO-устройства в просмотре сетевых соединений или просмотре устройств.
2. Перейти в свойствах интерфейса к пункту: Расширенные опции > Настройки реального времени > IO-цикл.
3. Выбрать необходимое число циклов из выпадающего списка.



11.2.15.3 Конфигурирование резервирования среды

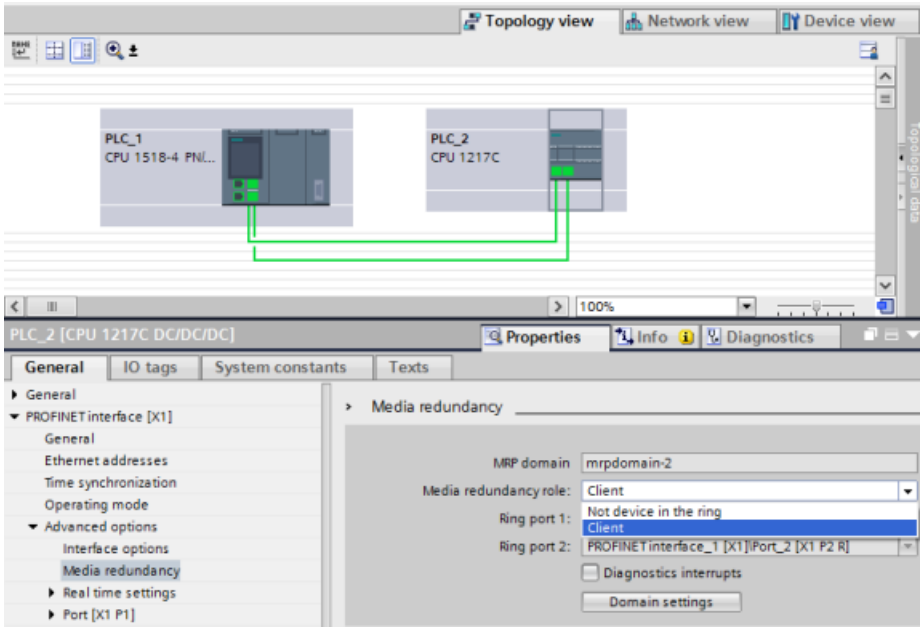
Все компоненты в приложении должны поддерживать протокол резервирования среды (MRP).

Порядок действий

Для конфигурирования резервирования среды выполнить следующие действия:

1. Создать кольцо, подключив порты соответствующим образом (напр., в просмотре топологии).
2. Выбрать устройство PROFINET, для которого необходимо сконфигурировать резервирование среды.

3. Перейти в окне инспектора к интерфейсу "PROFINET [X1] > Расширенные опции > Резервирование среды".



4. Назначить устройству в пункте "MRP-роль" одну из ролей "Менеджер (автом.)", "Клиент" или "Не является участником кольца".
   Если в просмотре топологии в TIA Portal конфигурируется кольцо, то TIA Portal определяет MRP-роль автоматически. Если устройство может быть менеджером, то TIA Portal выбирает "Менеджер (автом.)" в качестве MRP-роли. Для S7-1200 в качестве MRP-роли автоматический выбирается "Клиент".

   Примечание

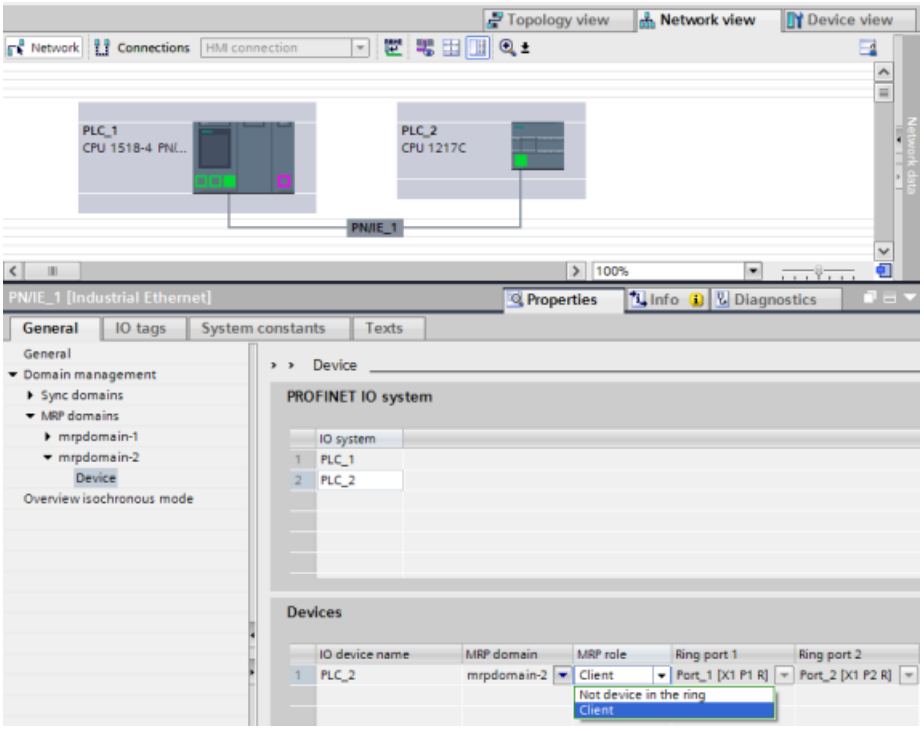
   Нельзя назначить S7-1200 CPU в качестве MRP-роли "Менеджер (автом.)".

5. Повторите шаги с 2 по 4 для всех устройств PROFINET в кольце.

Или:

1. Отметить систему PROFINET IO в просмотре сетевых соединений.
2. Кликнуть по системе PROFINET IO.

3. Перейти к устройству требуемого MRP-домена в окне инспектора.



4. Назначить устройствам PROFINET одну из ролей "Менеджер (автом.)", "Клиент" или "Не является участником кольца".

Примечание

Нельзя назначить S7-1200 CPU в качестве MRP-роли "Менеджер (автом.)".

Опция для "резервирования среды": MRP-роль

В зависимости от используемого устройства, доступны роли "Менеджер", "Менеджер (автом.)", "Клиент" и "Не является участником кольца".

Правила:

- В одном кольце может быть только одно устройство с ролью "Менеджер". Другие устройства с ролью "Менеджер" или "Менеджер (автом.)" запрещены. Все другие устройства в кольце могут быть только в роли "Клиента". Устройства, не входящие в кольцо, могут иметь роль "Не является участником кольца".
- Если в кольце отсутствует устройство с ролью "Менеджер", то в кольце должно быть минимум одно устройство с ролью "Менеджер (автом.)". В кольце может быть любое число устройств с ролями "Клиент" и "Менеджер (автом.)".

Примечание

Нельзя назначить S7-1200 CPU в качестве MRP-роли "Менеджер" или "Менеджер (автом.)".

Опция для "резервирования среды": Кольцевой порт 1 и кольцевой порт 2

Выбрать по отдельности порты, которые будут сконфигурированы как кольцевой порт 1 или кольцевой порт 2. В выпадающем списке представлены все возможные порты для каждого типа устройства. Если порты были установлены на заводе, то поля недоступны.

Примечание

Конфигурирование кольцевых портов для S7-1200 не требуется, так как S7-1200 CPU имеет только для порта.

Диагностические прерывания

Установить флажок "Диагностическое прерывание", если диагностические прерывания должны осуществляться для состояния MRP в локальном CPU. Могут быть сконфигурированы следующие диагностические прерывания:

- Ошибка подключения или порта:
  CPU создает диагностические прерывания для следующих ошибок на кольцевых портах:
  - Соседнее устройство кольцевого порта не поддерживает MRP.
  - Кольцевой порт соединен с не кольцевым портом.
  - Кольцевой порт соединен с кольцевым портом другого MRP-домена.
- Прерывание/возврат (только для MRP-менеджера):
  Если кольцо прерывается и выводится исходная конфигурация, CPU создает диагностические прерывания. Если интервал между двумя этими прерываниями не превышает 0,2 секунды, то это указывает на разрыв кольца.

На эти события можно реагировать в программе пользователя, запрограммировав соразмерную реакцию в OB диагностического прерывания (OB 82).

Примечание

Сторонние устройства в качестве MRP-менеджера

Для обеспечения безошибочной работы при использовании стороннего устройства в качестве MRP-менеджера в кольце, необходимо перед замыканием кольца назначить фиксированную роль "Клиент" всем остальным устройствам в кольце. В противном случае возможно зацикливание датаграмм и сбой сети.

11.2.16 S7-маршрутизация

Просмотр сетевых соединений в STEP 7 позволяет создавать сложные коммуникационные топологии, соединяя друг с другом устройства из разных S7 подсетей. Могут подключаться как классические S7-300/S7-400 CPU и CP, так и новейшие S7-CPU и CP, а также устройства HMI и рабочие станции, напр., OPC-сервер.

После принятия решения о том, какие устройства должны обмениваться данными, и установки необходимых соединений в STEP 7, система технических разработок (ES)

может загрузить соответствующие таблицы маршрутизации на различные маршрутизаторы S7 в рамках конфигурации оборудования. После загрузки таблиц маршрутизации на различные маршрутизаторы S7, ES или другие участники процесса коммуникации могут связаться с любым устройством, даже если устройства находятся в разных подсетях S7. Это возможно, потому что CPU и/или CP между ними действуют как маршрутизаторы S7. CPU и/или CP пересылают входящие запросы на соединение следующему маршрутизатору S7 до тех пор, пока запрос на соединение не достигнет целевого устройства, и устройства не установят соединение S7.

CPU использует механизм записи наборов данных для передачи таблиц маршрутизации, необходимых устройствам CP в локальном базовом устройстве. Таблицы маршрутизации устанавливают маршрут от одного устройства к другому во время запроса соединения, который включает в себя ID удаленной S7 подсети. Устройство, получающее запрос на соединение, опрашивает свою таблицу маршрутизации, находит следующую станцию на пути к целевой подсети S7 и направляет запрос на соединение дальше. В конце концов, запрос на соединение достигает намеченной цели, а ответ проходит тот же маршрут в обратном направлении.

S7-1200 CPU имеют один единственный интерфейс PN и до трех устройств CP, подключенных к локальной коммуникационной шине. Поэтому предлагается два варианта маршрутизации на станции S7-1200:

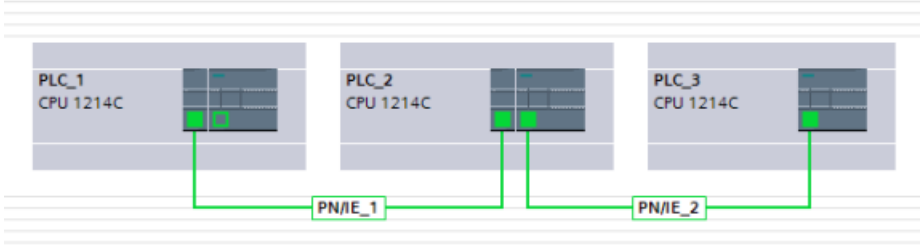
- Маршрутизация между CPU и CP
- Маршрутизация от одного CP к другому CP

Дополнительную информацию о всех S7-1200 CP, поддерживающих функцию "S7-маршрутизация", можно найти на странице технической поддержки веб-сайта онлайн-поддержки для промышленности Siemens. CP 1243-1 (https://support.industry.siemens.com/cs/us/en/view/584459) показан в качестве примера для CP. Выполнить поиск "S7-маршрутизации".

11.2.16.1 S7-маршрутизация между CPU и CP-интерфейсами

Поскольку S7-1200 CPU имеют одним интерфейс PN, автономный CPU не может играть роль маршрутизатора. Автономный CPU может быть одновременно подключен только к одной подсети S7. При установке модулей CP в локальном базовом устройстве CPU, можно подключиться к нескольким подсетям S7 и использовать маршрутизацию.

В приведенном ниже примере системы, для связи PLC_1 с PLC_3, система технических разработок (ES) должна направлять сообщения через PLC_2. ES должна загрузить таблицу маршрутизации для PLC_2, а PLC_2 должен предоставить таблицу маршрутизации для модуля CP в своем локальном базовом устройстве. С этими таблицами маршрутизации PLC_1 и PLC_3 могут связываться друг с другом, даже если они не подключены напрямую.

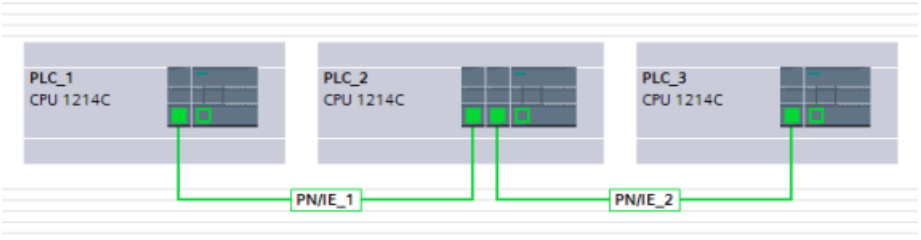


Для проверки маршрутизации из одной подсети S7 в другую подсеть S7, PLC_1 должен установить транспортное соединение с PLC_3, а PLC_3 должен установить соединение с PLC_1. При этом должна быть обеспечена возможность маршрутизации от интерфейса PN/IE контроллера к модулю CP, а также маршрутизация от модуля CP к интерфейсу PN/IE контроллера.

11.2.16.2 S7-маршрутизация между двумя интерфейсами CP

Поскольку S7-1200 CPU поддерживают до трех модулей CP, можно подключить все три модуля к различным подсетям S7. При установке как минимум двух модулей CP в локальном базовом устройстве CPU и подключении к различным подсетям S7, можно использовать маршрутизацию.

В приведенном ниже примере системы, для связи PLC_1 с PLC_3, система технических разработок (ES) должна направлять сообщения PLC_2 от модуля CP к модулю CP в локальном базовом устройстве. ES должна загрузить таблицу маршрутизации для PLC_2, а PLC_2 должен предоставить таблицу маршрутизации для двух модулей CP. С этими таблицами маршрутизации PLC_1 и PLC_3 могут связываться друг с другом, даже если они не подключены напрямую. Также необходимо учитывать, что маршрутизация от модуля CP к модулю CP выполняется без передачи сообщений через PN/IE интерфейс PLC_2.



11.2.17 Отключение SNMP

Простой протокол управления сетью (SNMP) - это стандартный Интернет-протокол для сбора и организации информации об управляемых устройствах в IP-сетях и для редактирования этой информации с целью изменения поведения устройства. Устройства с поддержкой SNMP включают в себя маршрутизаторы, коммутаторы, серверы, рабочие станции, принтеры, модемные стойки и многое другое.

SNMP широко используется в системах сетевого управления для мониторинга подключенных к сети устройств на наличие условий, требующих внимания администратора сети. SNMP использует различные сервисы и инструменты для обнаружения и диагностики топологии сети. Информация о свойствах устройств, поддерживающих SNMP, содержится в файлах базы информации управления (MIB), для которых пользователь должен иметь соответствующие права. SNMP предоставляет данные управления в виде переменных для управляемых систем, которые описывают конфигурацию системы. Эти переменные затем могут запрашиваться (и иногда определяться) управляющими приложениями.

SNMP использует транспортный протокол UDP и имеет два сетевых компонента:

- SNMP-менеджер: контролирует узлы сети.
- SNMP-клиент: Собирает различную специфическую для сети информацию в отдельных узлах сети и сохраняет ее в структурированной форме в базе управляющей информации (MIB). С этими данными может быть выполнена детальная диагностика сети.

При определенных условиях приложение может потребовать отключения SNMP. Примеры этих условий:

- Настройки системы безопасности в сети запрещают использование SNMP.
- У пользователя есть свое SNMP решение (напр., с собственными коммуникационными инструкциями).

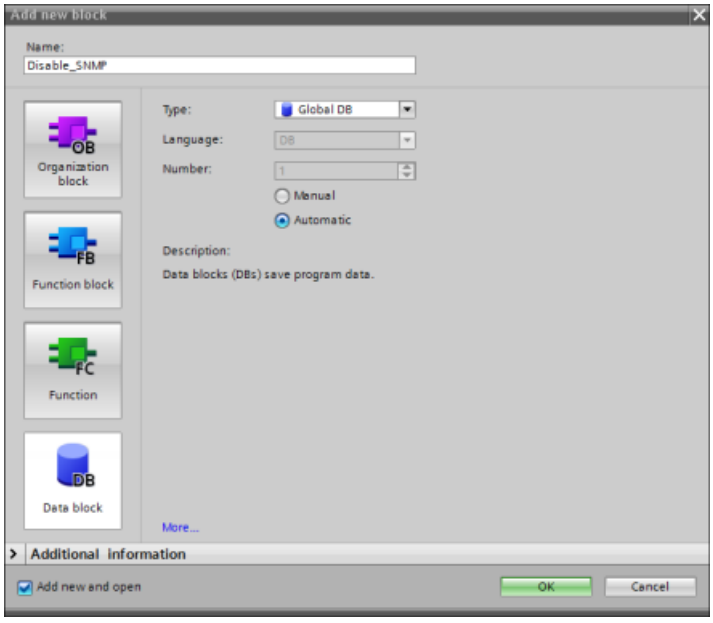
Если отключить SNMP для устройства, то станут недоступными некоторые опции для диагностики топологии сети (например, инструмент PRONETA или веб-сервер CPU).

11.2.17.1 Отключение SNMP

Отключение SNMP

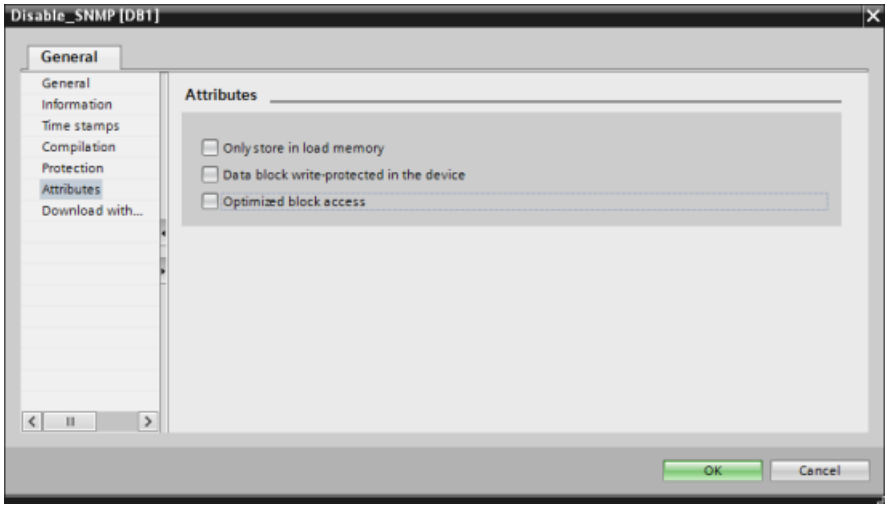
Для отключения SNMP в S7-1200 CPU выполнить следующие действия:

1. Создать классический блок данных (DB):



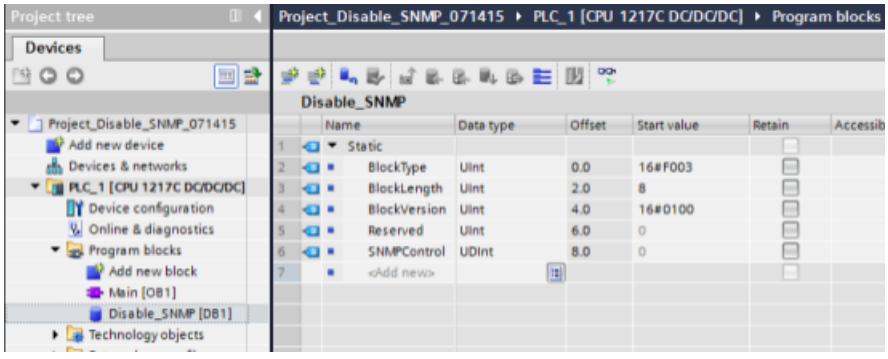
2. Выбрать свойства нового созданного DB.

3. Открыть вкладку "Атрибуты". Снять флажок "Оптимизированный доступ к блоку":

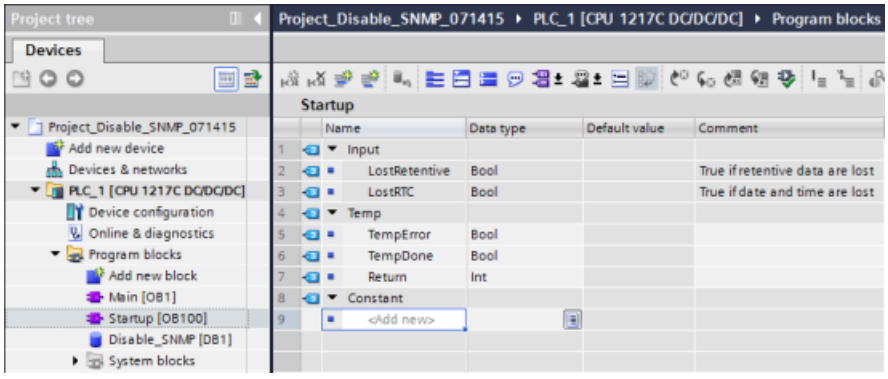


4. Нажать кнопку "ОК".
   После появится сообщение с рекомендацией о повторной компиляции программы. Выполнить повторную компиляцию программы.

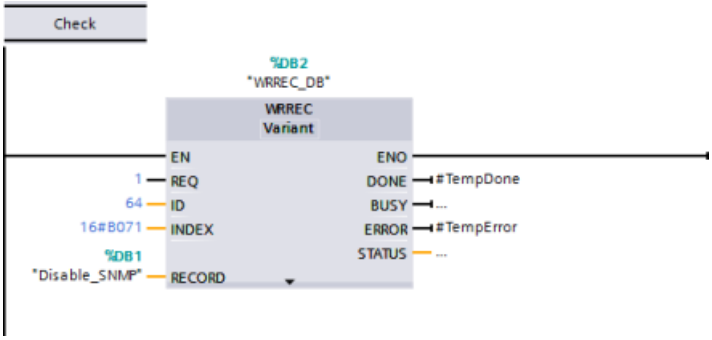
5. В классическом блочном интерфейсе DB создать следующие статические переменные (теги) с указанными значениями. Эти переменные (теги) будут использоваться в программе для отключения внутренней реализации SNMP:



6. Добавить в пусковой ОВ (OB100) временные переменные, как показано ниже:



7. Используя редактор LAD, вставить в пусковой OB (OB100) в сети 1 инструкцию Label (метка перехода) (в приведенном ниже примере метка перехода называется "Check") и инструкцию WRREC (запись набора данных) с показанными входами и выходами:



8. Вставить следующий цикл и следующий контрольный код с выходом "Переход на метку" (JMP). Этот код обеспечивает завершение вызова и отключение SNMP до выхода из пускового OB:



11.2.18 Диагностика

Информацию об использовании организационных блоков (OB) для диагностики с этими коммуникационными сетями, можно найти в разделе "Организационные блоки (OB)" (Страница 91).

11.2.19 Инструкции для распределенной периферии

Дополнительную информацию об использовании инструкции для распределенной периферии с этими коммуникационными сетями можно найти в разделе "Распределенная периферия (PROFINET, PROFIBUS или AS-i)" (Страница 386).

11.2.20 Диагностические инструкции

Дополнительную информацию об использовании этих инструкции с этими коммуникационными сетями можно найти в разделе "Диагностика (PROFINET или PROFIBUS): диагностические инструкции" (Страница 439).

11.2.21 Диагностические события для распределенной периферии

Дополнительную информацию об использовании этой диагностической информации с этими коммуникационными сетями можно найти в разделе "Диагностика (PROFINET или PROFIBUS): диагностические события для распределенной периферии" (Страница 486).

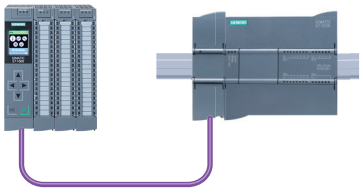
11.3 PROFIBUS

Система PROFIBUS использует задатчик шины для опроса ведомых устройств, подключенных в разных местах к последовательной шине RS485. Ведомое устройство PROFIBUS - это любое периферийное устройство (преобразователь ввода-вывода, вентиль, привод двигателя или измерительное устройство), которое обрабатывает данные и отправляет информацию ведущему устройству. Ведомое устройство представляет собой пассивную станцию в сети, поскольку оно не имеет прав доступа к шине. Оно может только подтверждать полученные сообщения или отправлять ответные сообщения ведущему устройству по запросу. Все ведомые устройства PROFIBUS имеют одинаковый приоритет, и вся сетевая коммуникация зависит от ведущего устройства.

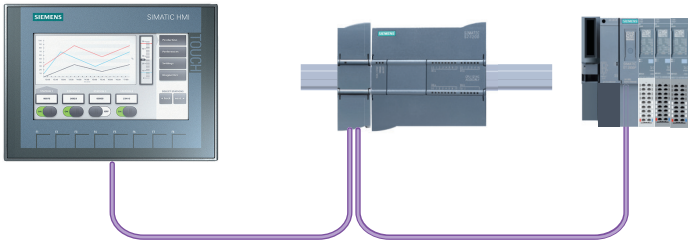
Ведущее устройство PROFIBUS представляет собой "активную станцию" в сети. PROFIBUS DP определяет два класса ведущих устройств. Ведущее устройство класса 1 (обычно центральный программируемый контроллер (PLC) или PC, на котором работает специальное программное обеспечение), обрабатывает обычную коммуникацию или обмен данными с назначенными ему ведомыми устройствами. Ведущее устройство класса 2 (обычно устройство конфигурирования, напр., ноутбук или пульт программирования для ввода в эксплуатацию, обслуживания или диагностики) является специальным устройством, используемым прежде всего для ввода в эксплуатацию ведомых устройств и для диагностики.

S7-1200 подключается к сети PROFIBUS как ведомое устройство DP с помощью коммуникационного модуля CM 1242-5. Модуль CM 1242-5 (ведомое устройство DP) может быть коммуникационным партнером ведущих устройств DP V0/V1. Необходимый для конфигурирования модуля в сторонней системе GSD-файл для CM 1242-5 (ведомое устройство DP) можно найти на CD, прилагаемом к модулю, и в Интернете на странице поддержки клиентов автоматизации (https://support.industry.siemens.com/cs/ww/en/ps/6GK7242-5DX30-0XE0) Siemens.

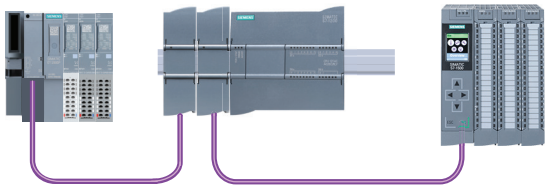
На рисунке ниже, S7-1200 является ведомым DP устройством контроллера S7-300:



S7-1200 подключен к сети PROFIBUS как ведущее устройство DP с помощью коммуникационного модуля CM 1243-5. Модуль CM 1243-5 (ведущее устройство DP) может быть коммуникационным партнером ведомых устройств DP V0/V1. На рисунке ниже S7-1200 является ведущим устройством, управляющим ведомым DP устройством ET200SP:



Если CM 1242-5 и CM 1243-5 установлены вместе, то S7-1200 может одновременно выступать и как ведомое устройство ведущей DP-системы верхнего уровня, и как ведущее устройство ведомой DP-системы нижнего уровня.



В V3.0 можно сконфигурировать максимум три PROFIBUS-CM на станцию, среди которых может быть любая комбинация CM ведущих и ведомых DP устройств. Ведущие устройства DP в V3.0 или более поздней версии прошивки для CPU могут управлять максимумом 32 ведомыми устройствами каждый.

Данные конфигурации PROFIBUS-CM сохраняются на локальном CPU. Это обеспечивает простую замену этих коммуникационных модулей при необходимости.

Чтобы использовать PROFIBUS с S7-1200 CPU от версии V4.0, необходимо обновить прошивку для CM ведущего устройства PROFIBUS минимум до версии V1.3.

Примечание

Всегда следует использовать только последнюю версию прошивки PROFIBUS CM (http://support.automation.siemens.com/WW/view/en/42131407). Обновление прошивки может быть выполнено одним из следующих способов:

- С помощью Интерактивных и диагностических инструментов STEP 7 (Страница 1394)
- С помощью карты памяти SIMATIC (Страница 147)
- Через стандартную веб-страницу "Информация о модулях" веб-сервера (Страница 1046)
- С помощью SIMATIC Automation Tool (https://support.industry.siemens.com/cs/ww/en/view/98161300)

11.3.1 Коммуникационные службы PROFIBUS-CM

Модули PROFIBUS-CM используют протокол PROFIBUS DP-V1.

Типы коммуникации для DP-V1

Следующие типы коммуникации доступны для DP-V1:

- Циклическая коммуникация (CM 1242-5 и CM 1243-5)
  Оба PROFIBUS модуля поддерживают циклическую коммуникацию для передачи данных процесса между ведомым и ведущим DP устройствами.
  Циклическая коммуникация обрабатывается операционной системой CPU. Программные блоки для этого не нужны. Данные ввода-вывода считываются или записываются непосредственно из / в образ процесса CPU.
- Ациклическая коммуникация (только CM 1243-5)
  Ведущее устройство DP также поддерживает ациклическую коммуникацию с использованием программных блоков:
  - Инструкция "RALRM" доступна для обработки прерываний.
  - Инструкции "RDREC" и "WRREC" доступны для передачи диагностических и конфигурационных данных.

Не поддерживаемые CM 1243-5 функции: SYNC/FREEZE и Get_Master_Diag

Другие коммуникационные службы CM 1243-5

Модуль ведущего DP устройства CM 1243-5 поддерживает следующие дополнительные коммуникационные службы:

- Коммуникация S7
  - PUT/GET службы
    Ведущее устройство DP функционирует как клиент и сервер для запросов от других S7 контроллеров или PC по PROFIBUS.
  - Коммуникация PG/OP
    Функции PG обеспечивают загрузку конфигурационных данных и пользовательских программ из программатора и передачу диагностических данных в программатор.
    Возможными участниками процесса для OP-коммуникаций являются HMI-панели, панельные PC SIMATIC WinCC flexible или SCADA-системы, которые поддерживают S7-коммуникацию.

11.3.2 Ссылка на руководства пользователя для PROFIBUS-CM

Дополнительная информация

Подробную информацию о PROFIBUS-CM можно найти в справочниках/руководствах по соответствующим устройствам. Они находятся в Интернете на страницах Поддержки клиентов Автоматизации производства от Siemens и имеют следующие ID:

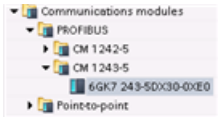
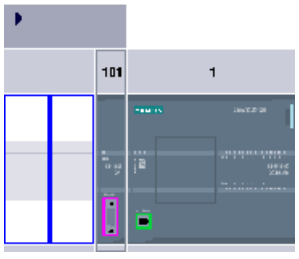
- CM 1242-5 (https://support.industry.siemens.com/cs/ww/en/ps/15667)
- CM 1243-5 (https://support.industry.siemens.com/cs/ww/en/ps/15669)

11.3.3 Конфигурирование ведущего и ведомого устройства DP

11.3.3.1 Добавление CM 1243-5 (ведущее устройство DP) и ведомого устройства DP

Следует использовать каталог оборудования на портале "Устройства и сети" для добавления модулей PROFIBUS к CPU. Эти модули подключаются с левой стороны от модуля CPU. Для вставки модуля в конфигурацию устройства, выбрать его в каталоге оборудования и дважды кликнуть по нему или перетащить модуль в отмеченный слот.

Таблица 11-60 Добавление PROFIBUS CM 1243-5 (ведущее устройство DP) в конфигурацию устройства

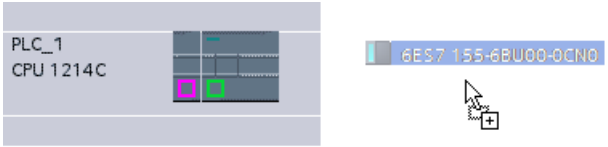
Модуль	Выбор модуля	Вставка модуля	Результат
CM 1243-5 (ведущее устройство DP)			

Можно использовать каталог оборудования для добавления и ведомых устройств DP. Например, для добавления ET200 SP ведомого DP устройства следует открыть в каталоге оборудования следующие папки:

- Распределенный ввод-вывод
- ET 200SP
- Интерфейсные модули
- PROFIBUS

После выбрать "6ES7 155-6BU00-0CN0" (IM155-6 DP HF) из списка заказных номеров и вставить ET200SP ведомое устройство DP как показано на рисунке ниже.

Таблица 11-61 Добавление ET200SP ведомого устройства DP в конфигурацию устройства

Вставка ведомого устройства DP	Результат
	

11.3.3.2 Конфигурирование логических сетевых соединений между двумя устройствами PROFIBUS.

После конфигурирования модуля CM 1243-5 (ведущее устройство DP), можно сконфигурировать свои сетевые соединения.

На портале "Устройства и сети" в "Просмотре сетевых соединений" можно объединять в сеть устройства в проекте. Чтобы создать PROFIBUS соединение, выбрать фиолетовое поле (PROFIBUS) на первом устройстве. Перетащить линию к полю PROFIBUS на втором устройстве. Отпустить кнопку мыши. Соединение PROFIBUS создано.

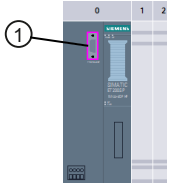
Подробную информацию см. в "Конфигурация устройства: создание сетевого соединения" (Страница 811).

11.3.3.3 Назначение адресов PROFIBUS для CM 1243-5 и ведомого устройства DP

Конфигурирование интерфейса PROFIBUS

После конфигурирования логических сетевых соединений между двумя устройствами PROFIBUS, можно сконфигурировать параметры для интерфейсов PROFIBUS. Для этого кликнуть по фиолетовому полю PROFIBUS на модуле CM 1243-5. На вкладке "Свойства" в окне инспектора отобразится интерфейс PROFIBUS. Интерфейс PROFIBUS ведомого устройства DP конфигурируется аналогичным образом.

Таблица 11-62 Конфигурирование PROFIBUS-интерфейсов модуля CM 1243-5 (ведущее устройство DP) и ведомого DP устройства ET200SP

CM 1243-5 (ведущее устройство DP)	ET 200SP ведомое устройство DP
	

① Порт PROFIBUS

Назначение адреса PROFIBUS

В сети PROFIBUS каждому устройству назначается адрес PROFIBUS. Этот адрес может находиться в диапазоне от 0 до 127, принимая во внимание следующие исключения:

- Адрес 0: Зарезервирован для конфигурации сети и/или инструментов программирования, подключенных к шине
- Адрес 1: Зарезервирован Siemens для первого ведущего устройства

- Адрес 126: Зарезервирован для устройств от изготовителя, не имеющих установочных переключателей, адреса которых должны быть переназначены через сеть
- Адрес 127: Зарезервирован для широковещательных сообщений ко всем устройствам в сети и не может быть назначен работоспособным устройствам

Таким образом, для работоспособных устройств PROFIBUS могут использоваться адреса от 2 до 125.

Выбрать в окне свойств запись "Адрес PROFIBUS". STEP 7 отображает диалоговое окно конфигурирования для PROFIBUS-адреса, которое используется, чтобы назначить PROFIBUS-адрес устройства.

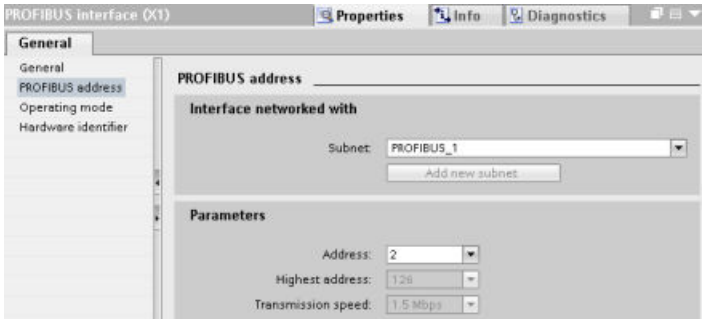


Таблица 11-63 Параметры для адреса PROFIBUS

Параметр		**Описание**
Подсеть	Имя подсети, к которой подключено устройство. Нажать кнопку "Добавить новую подсеть", чтобы создать новую подсеть. Установкой по умолчанию является "Не соединено". Возможны два типа соединений: • Установка по умолчанию "Не соединено" предоставляет локальное соединение. • Подсеть необходима, если сеть содержит два или более устройств.	
Параметр	Адрес	Назначенный устройству PROFIBUS-адрес
	Максимальный адрес	Максимальный PROFIBUS-адрес основывается на активных станциях на PROFIBUS (например, ведущее устройство DP). Пассивные ведомые устройств DP имеют независимые PROFIBUS-адреса от 1 до 125, даже если максимальный PROFIBUS-адрес установлен, к примеру, на 15. Максимальный PROFIBUS-адрес важен для передачи маркера (перенаправление прав на передачу), и маркер передается только активным станциям. Определение наибольшего PROFIBUS-адреса оптимизирует шину.
	Скорость передачи данных	Скорость передачи данных в сконфигурированной сети PROFIBUS: Скорость передачи для PROFIBUS может лежать в диапазоне от 9,6 кбит/с до 12 Мбит/с. Она зависит от свойств используемого узла PROFIBUS. Скорость передачи не должна быть больше скорости самого медленного узла. Скорость передачи обычно устанавливается для ведущего устройства в сети PROFIBUS, при этом все ведомые устройства DP автоматически используют ту же самую скорость передачи (автобод).

11.3.4 Инструкции для распределенной периферии

Дополнительную информацию об использовании инструкции для распределенной периферии с этими коммуникационными сетями можно найти в разделе "Распределенная периферия (PROFINET, PROFIBUS или AS-i)" (Страница 386).

11.3.5 Диагностические инструкции

Дополнительную информацию об использовании этих инструкции с этими коммуникационными сетями можно найти в разделе "Диагностика (PROFINET или PROFIBUS): диагностические инструкции" (Страница 439).

11.3.6 Диагностические события для распределенной периферии

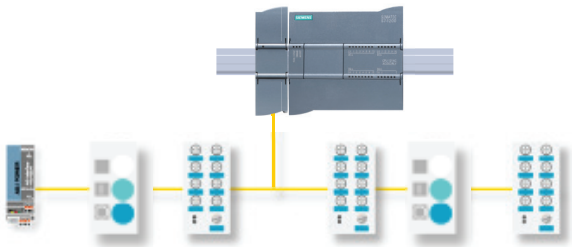
Дополнительную информацию об использовании этой диагностической информации с этими коммуникационными сетями можно найти в разделе "Диагностика (PROFINET или PROFIBUS): диагностические события для распределенной периферии" (Страница 486).

11.4 ASi

S7-1200 CM 1243-2 ведущее устройство AS-i обеспечивает подключение S7-1200 CPU к сети AS-i.

Интерфейс исполнительного устройства/датчика или AS-i является системой подключения к сети для ведущего устройства на самом нижнем уровне в системах автоматизации. CM 1243-2 служит ведущим устройством AS-i в сети. Используя простой AS-i кабель, датчики и исполнительные устройства (ведомые устройства AS-i) могут быть подключены к CPU через CM 1243-2. CM 1243-2 осуществляет всю координацию сети AS-i и перенаправляет данные и информация о статусе от приводов и исполнительных устройств в CPU через адреса ввода-вывода, назначенные CM 1243-2. В зависимости от типа ведомого устройства, возможен доступ к двоичным или аналоговым значениям. Ведомые устройства AS-i представляют собой каналы ввода и вывода системы AS-i и активны только в том случае, если вызываются CM 1243-2.

На рисунке ниже S7-1200 представляет собой ведущее AS-i устройство управляющее AS-i модульными ведомыми устройствами цифрового/аналогового ввода/вывода.



Чтобы использовать AS-i с S7-1200 CPU версии V4.0, необходимо обновить прошивку для CM ведущего устройства до версии V1.1.

Для выполнения обновления можно использовать веб-сервер или карту памяти SIMATIC.

Примечание

Для S7-1200 CPU версии 4.0 при использовании веб-сервера или карты памяти SIMATIC для апгрейда прошивки AS-i от версии V1.0 до версии V1.1 необходимо обновить прошивку AS-i в ведущем AS-i устройстве CM 1243-2 следующим образом:

1. Загрузить апгрейд прошивки в ведущее AS-i устройство CM 1243-2.
2. Когда загрузка будет завершена, выполнить выключения и включения питания S7-1200 CPU, чтобы полностью завершить процесс апгрейда прошивки в ведущем AS-i устройстве CM 1243-2.
3. Повторить шаги 1 и 2 для каждого дополнительного ведущего AS-i устройства CM 1243-2. S7-1200 PLC позволяет подключить максимум три ведущих AS-i устройства CM 1243-2.

Примечание

Всегда следует использовать только последнюю версию прошивки AS-i CM (http://support.automation.siemens.com/WW/view/en/43416171). Актуальную версию прошивки можно найти на Интернет-сайте сервиса и поддержки Siemens.

11.4.1 Конфигурирование ведущего и ведомого AS-i устройств

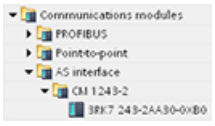
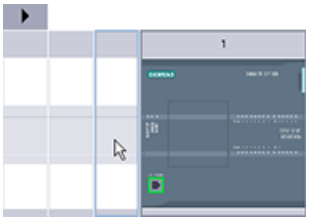
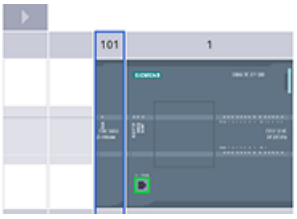
Ведущее AS-i устройство CM 1243-2 интегрируется в систему автоматизации S7-1200 как коммуникационный модуль.

Подробную информацию о ведущем AS-i устройстве CM 1243-2 можно найти в руководстве "Ведущее AS-i устройство CM 1243-2 и AS-i разделительный модуль данных DCM 1271 для SIMATIC S7-1200" (https://support.industry.siemens.com/cs/ww/en/ps/15750/man).

11.4.1.1 Добавление модуля ведущего AS-i устройства CM 1243-2 и ведомого AS-i устройства

Можно использовать каталог оборудования для добавления модулей ведущего AS-i устройства CM1243-2 к CPU. Эти модули подключаются с левой стороны от модуля CPU. Можно использовать до трех модулей ведущего AS-i устройства CM1243-2. Для вставки модуля в конфигурацию устройства, выбрать его в каталоге оборудования и дважды кликнуть по нему или перетащить модуль в отмеченный слот.

Таблица 11-64 Добавление модуля ведущего AS-i устройства CM1243-2 в конфигурацию устройства

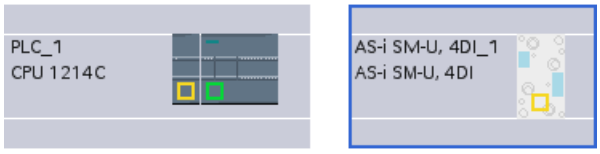
Модуль	Выбор модуля	Вставка модуля	Результат
CM 1243-2 ведущее AS-i устройство			

Можно использовать каталог оборудования для добавления и ведомых устройств AS-i. Например, для добавления ведомого устройства с параметрами "модуль ввода-вывода, компактный цифровой ввод" следует открыть в каталоге оборудования следующие контейнеры:

- Полевые устройства
- Ведомые AS-Interface устройства

Затем выбрать "3RG9 001-0AA00" (AS-i SM-U, 4DI) из списка заказных номеров и добавить ведомое устройство "модуль ввода-вывода, компактный цифровой ввод", как показано на рисунке ниже.

Таблица 11-65 Добавление ведомого AS-i устройства в конфигурацию устройства

Вставка ведомого AS-i устройства	Результат
	

11.4.1.2 Конфигурирование логических сетевых соединений между двумя устройствами AS-i.

После конфигурирования ведущего AS-i устройства CM1243-2, можно сконфигурировать свои сетевые соединения.

На портале "Устройства и сети" в "Просмотре сетевых соединений" можно объединять в сеть устройства в проекте. Чтобы создать AS-i соединение, выбрать желтое поле (AS-i) на

первом устройстве. Перетащить линию к полю AS-i на втором устройстве. Отпустить кнопку мыши. Соединение AS-i создано.

Дополнительную информацию см. в Конфигурация устройства: создание сетевого соединения" (Страница 811).

11.4.1.3 Конфигурирование свойств ведущего AS-i устройства CM1243-2

Чтобы сконфигурировать параметры для AS-i интерфейса, кликнуть по желтому AS-i полю на модуле ведущего AS-i устройства CM1243-2. На вкладке "Свойства" в окне инспектора отобразится интерфейс AS-i.

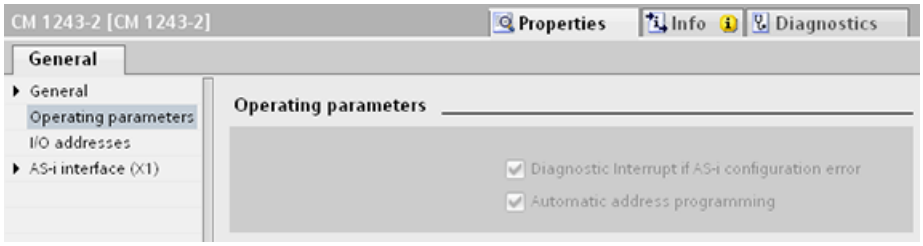
В окне инспектора STEP 7 можно просмотреть, сконфигурировать и изменить общую информацию, адреса и рабочие параметры:

Таблица 11-66 Свойства модуля ведущего AS-i устройства CM1243-2

Свойство	Описание
Общее	Имя ведущего AS-i устройства CM 1243-2
Рабочие параметры	Параметры для ответа ведущего AS-i устройства
Адреса ввода-вывода	Адресное пространство для адресов ввода-вывода ведомых устройств
AS-i интерфейс (X1)	Назначенная сеть AS-i

Примечание

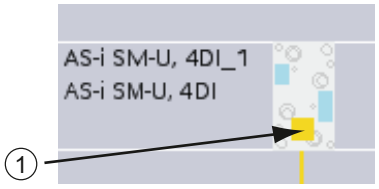
Опции "Диагностическое прерывание для ошибок в конфигурации AS-i" и "Автоматическое программирование адресов" активны всегда и поэтому недоступны для выбора (серые).



11.4.1.4 Назначение AS-i адреса ведомому AS-i устройству

Конфигурирование интерфейса ведомого AS-i устройства

Чтобы сконфигурировать параметры для AS-i интерфейса, кликнуть по желтому AS-i полю на ведомом AS-i устройстве. На вкладке "Свойства" в окне инспектора отобразится интерфейс AS-i.



1 Порт AS-i

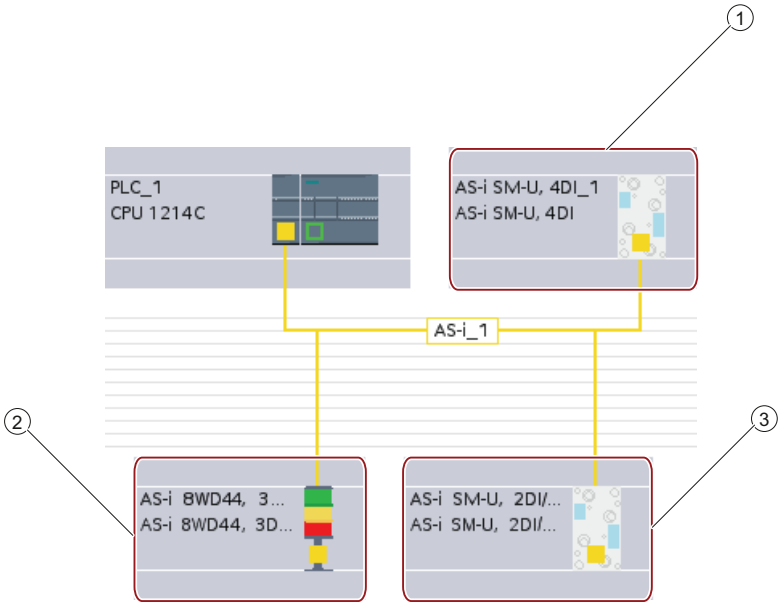
Назначение адреса ведомому AS-i устройству

В сети AS-i каждому устройству назначается адрес ведомого устройства AS-i. Этот адрес может находиться в диапазоне от 0 до 31. Адрес 0 зарезервирован только для новых ведомых устройств. Доступны адреса от 1 (A или B) до 31 (A или B) максимум для 62 ведомых устройств.

"Стандартные" AS-i устройства используют полный числовой адрес без идентификатора A или B. AS-i устройства типа "A/B узла" используют A или B часть каждого адреса соответственно, поэтому каждый из 31 адресов может использоваться дважды. Диапазон адресов распространяется от 1A до 31A плюс от 1B до 31B.

Любой адрес в диапазоне от 1 до 31 может быть назначен ведомому AS-i устройству. Другими словами: Не имеет значения, начинаются ли ведомые устройства с адреса 21 или первому ведомому устройству в действительности назначен адрес 1.

В примере ниже, три AS-i устройства адресованы как "1" (устройство стандартного типа), "2 A" (устройство типа узла A/B), и "3" (устройство стандартного типа):

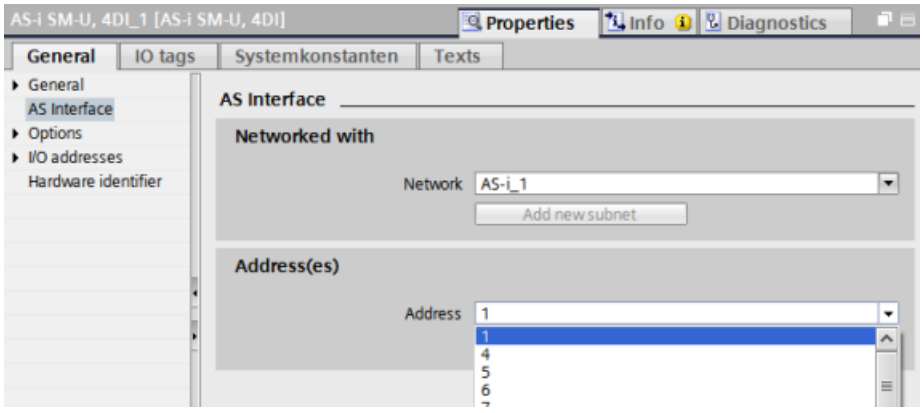


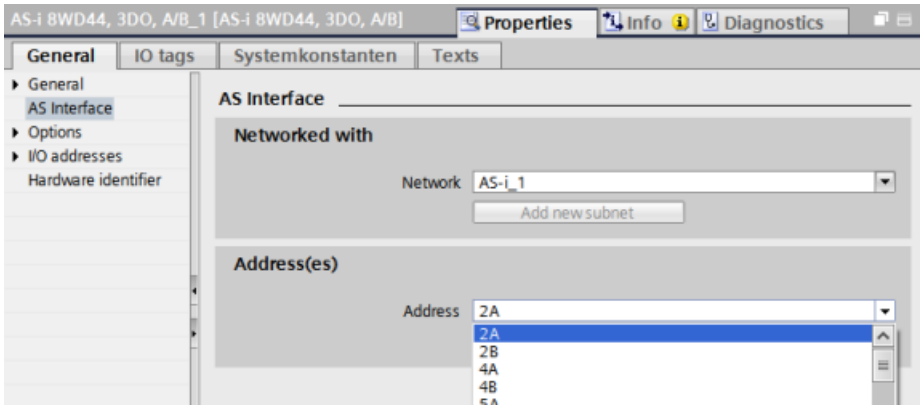
① Адрес ведомого AS-i устройства 1; устройство: AS-i SM-U, 4DI; заказной №: 3RG9 001-0AA00

② Адрес ведомого AS-i устройства 2A; устройство: AS-i 8WD44, 3DO, A/B; заказной №: 8WD4 428-0BD

③ Адрес ведомого AS-i устройства 3; устройство: AS-i SM-U, 2DI/2DO; заказной №: 3RG9 001-0AC00

Здесь следует ввести адрес ведомого AS-i устройства:





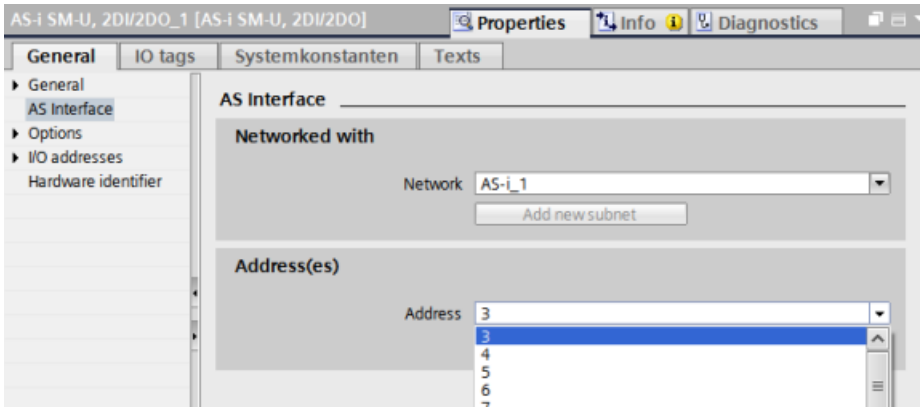


Таблица 11-67 Параметры для AS-i интерфейса

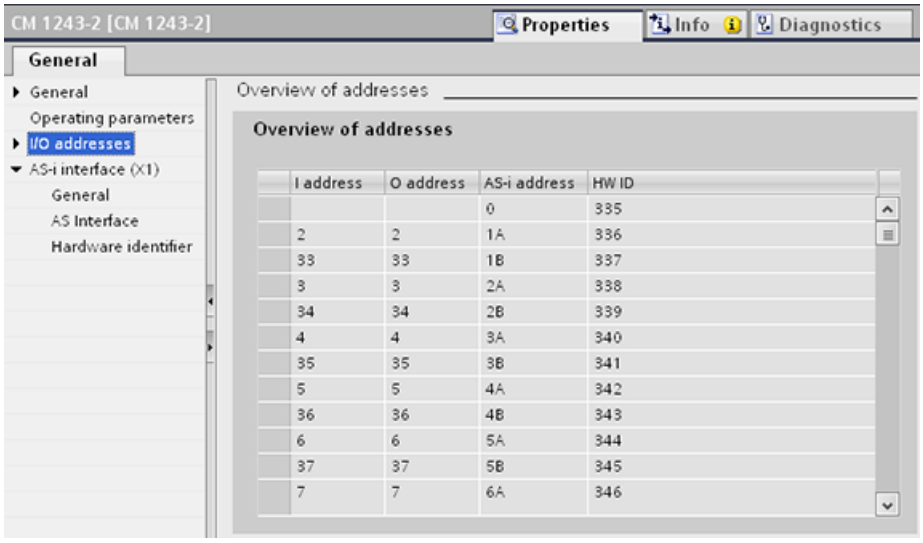
Параметр	Описание
Сеть	Имя сети, к которой подключено устройство.
Адрес(а)	Назначенный AS-i адрес для ведомого устройства в диапазоне от 1 (A или B) до 31 (A или B) в общей сложности максимум для 62 ведомых устройств.

11.4.2 Обмен данными между программой пользователя и ведомыми AS-i устройствами

11.4.2.1 Базовая конфигурация в STEP 7

Ведущее AS-i устройство резервирует 62-байтовую область данных в области ввода-вывода CPU. Доступ к цифровым данным выполняется здесь в байтах. Для каждого ведомого устройства есть один байт входных и один байт выходных данных

Назначение AS-i соединений цифровых ведомых AS-i устройств битам данных назначенного байта отображается в окне инспектора модуля ведущего AS-i устройства CM 1243-2.



Можно обращаться к данным ведомых AS-i устройств в программе пользователя, используя отображаемые адреса ввода-вывода с соответствующими битовыми логическими операциями (например, "И") или назначением битов.

Примечание

"Назначение системой" активируется автоматически, если для конфигурирования ведомых AS-i устройств не используется STEP 7

Если ведомые устройства не конфигурируются, то необходимо сообщить ведущему AS-i устройству CM1243-2 о фактической конфигурации шины. Для этого используется онлайновая функция "ФАКТИЧЕСКАЯ > ЗАДАННАЯ".

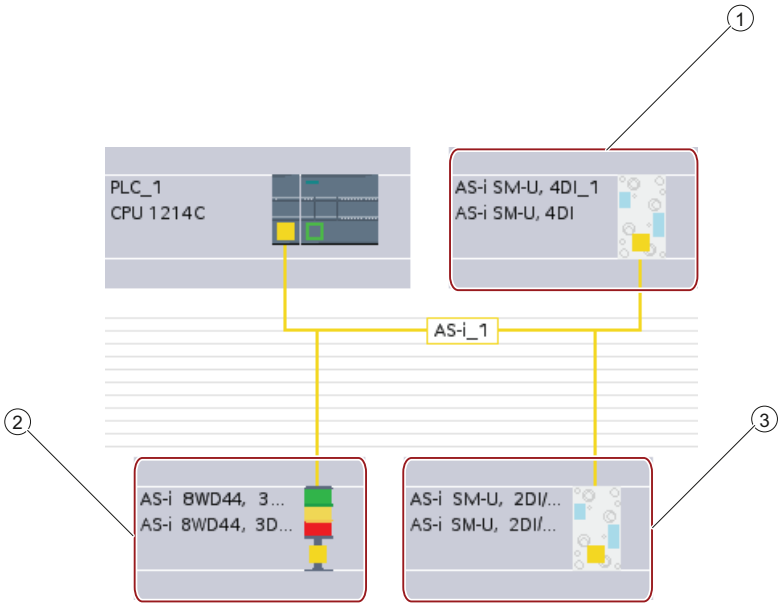
Дополнительная информация

Подробную информацию о ведущем AS-i устройстве CM 1243-2 можно найти в руководстве "Ведущее AS-i устройство CM 1243-2 и AS-i разделительный модуль данных DCM 1271 для SIMATIC S7-1200" (https://support.industry.siemens.com/cs/ww/en/ps/15750/man).

11.4.2.2 Конфигурирование ведомых устройств с помощью STEP 7

Передача цифровых AS-i значений

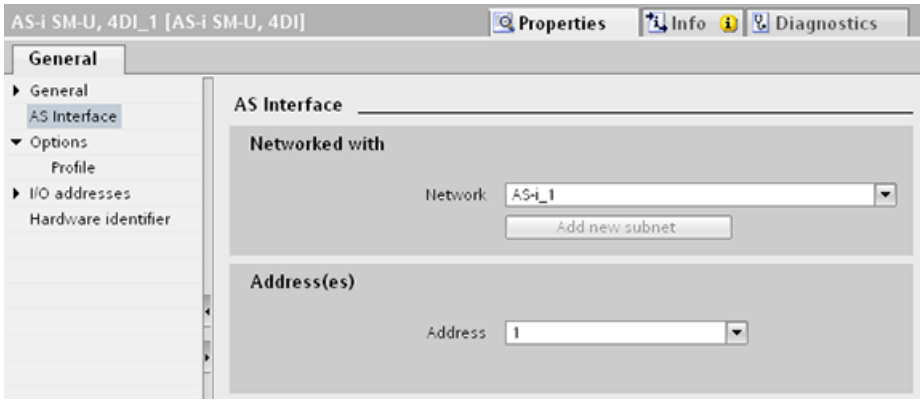
В циклическом режиме CPU обращается к цифровым входам и выходам ведомых AS-i устройств через ведущее AS-i устройство CM1243-2. Доступ к данным осуществляется через адреса ввода-вывода или посредством передачи наборов данных



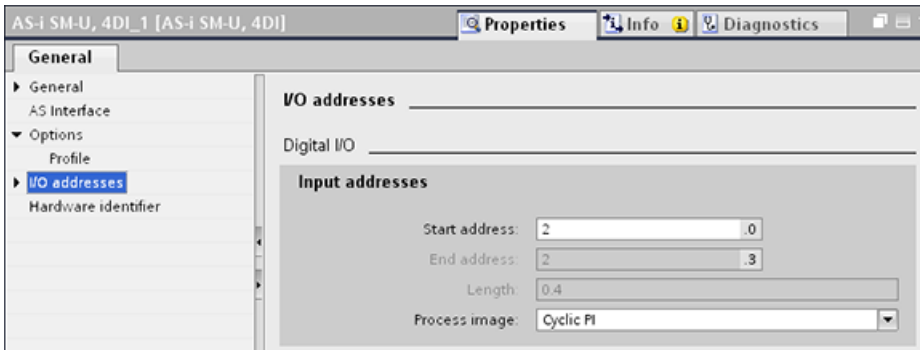
① Адрес ведомого AS-i устройства 1
② Адрес ведомого AS-i устройства 2A
③ Адрес ведомого AS-i устройства 3

Доступ к цифровым данным выполняется здесь в байтах (другими словами: один байт назначен каждому AS-i цифровому ведомому устройству). При конфигурировании ведомых AS-i устройств в STEP 7, адрес ввода-вывода для доступа к данным из пользовательской программы отображается в окне инспектора для соответствующего ведомого AS-i устройства.

Модулю ввода цифровых сигналов (AS-i SM-U, 4DI) в AS-i сети, показанной выше, был назначен адрес ведомого устройства 1. Клик по модулю ввода цифровых сигналов, вкладка "AS interface" в "Свойствах" устройства отобразит адрес ведомого устройства, как показано ниже:

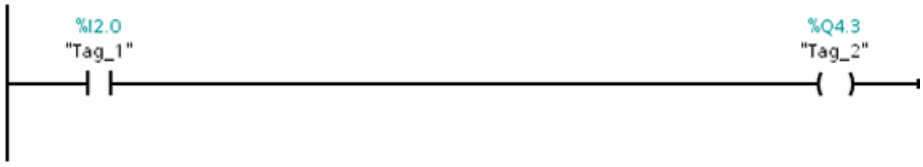


Модулю ввода цифровых сигналов (AS-i SM-U, 4DI) в AS-i сети, показанной выше, был назначен адрес ввода-вывода 2. Клик по модулю ввода цифровых сигналов, вкладка "Адрес ввода-вывода" в "Свойствах" устройства отобразит адрес ввода-вывода, как показано ниже:



Можно обращаться к данным ведомых AS-i устройств в программе пользователя, используя их адреса ввода-вывода с соответствующими битовыми логическими операциями (например, "И") или назначением битов. Следующая простая программа показывает принцип назначения:

В этой программе опрашивается вход 2.0. В AS-i системе этот вход относится к ведомому устройству 1 (входной байт 2, бит 0). Выход 4.3, который затем устанавливается, соответствует ведомому AS-i устройству 3 (выходной байт 4, бит 3).



Передача аналоговых AS-i значений

Можно обращаться к аналоговым данным ведомого AS-i устройства через образ процесса CPU, если это ведомое AS-i устройство было сконфигурировано в STEP 7 как аналоговое ведомое устройство.

Если аналоговое ведомое устройство не было сконфигурировано в STEP 7, то обращение к данным ведомого AS-i устройства возможно только через ациклические функции

(интерфейс набора данных). В пользовательской программе CPU вызовы AS-i считываются и записываются через инструкции для распределенной периферии, а именно RDREC (чтение набора данных), и WRREC (запись набора данных).

Примечание

Указанная через STEP 7 и загруженная на станцию S7 конфигурация ведомых AS-i устройств, передается из CPU в ведущее AS-i устройство CM1243-2 во время запуска станции S7. Любая возможно существующая конфигурация, которая была определена через онлайн-функцию "Назначение системой" (Страница 996) ("ФАКТИЧЕСКАЯ-> ЗАДАННАЯ"), будет перезаписана.

Дополнительная информация

Подробную информацию о ведущем AS-i устройстве CM 1243-2 можно найти в руководстве "Ведущее AS-i устройство CM 1243-2 и AS-i разделительный модуль данных DCM 1271 для SIMATIC S7-1200" (http://support.automation.siemens.com/WW/view/en/50414115/133300).

11.4.3 Инструкции для распределенной периферии

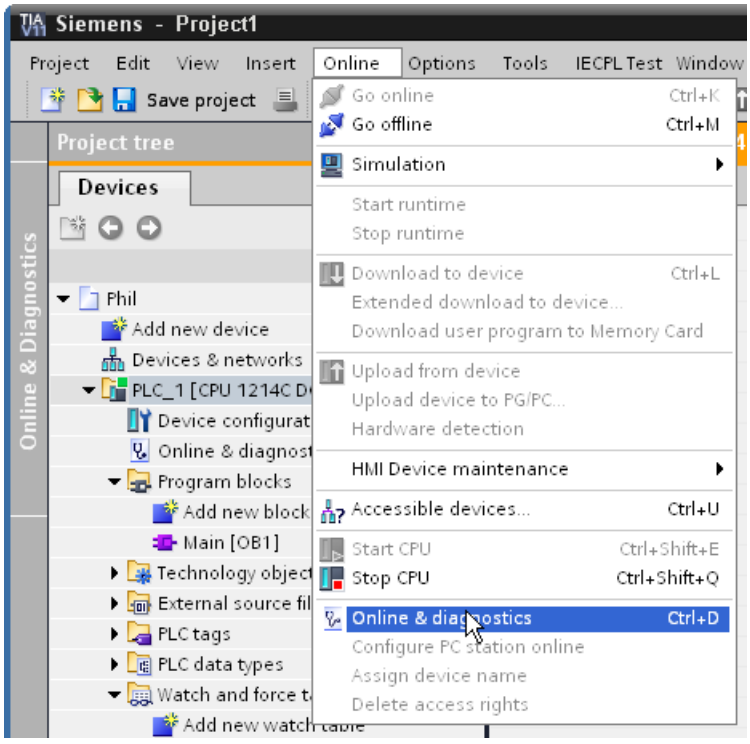
Дополнительную информацию об использовании инструкции для распределенной периферии с этими коммуникационными сетями можно найти в разделе "Распределенная периферия (PROFINET, PROFIBUS или AS-i)" (Страница 386).

11.4.4 Работа с онлайн-инструментами AS-i

Изменение режимов работы AS-i онлайн

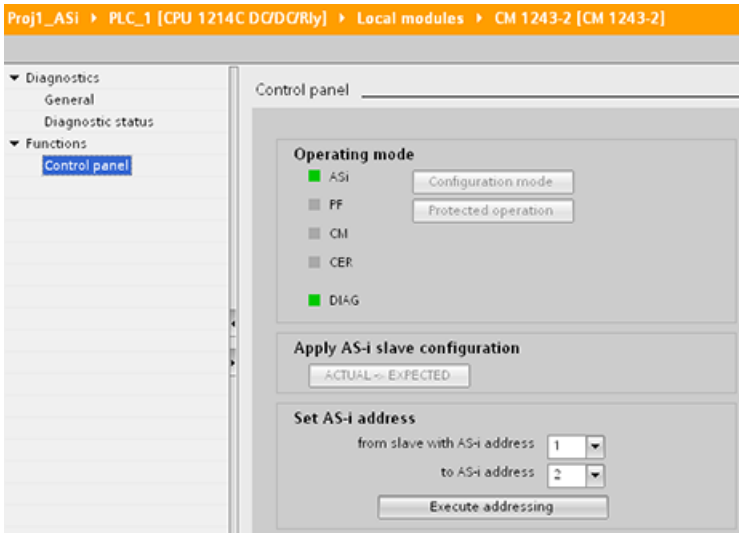
Для отображения и изменения режимов работы AS-i необходимо перейти в режим онлайн.

Для перехода в режим онлайн, сначала необходимо выбрать модуль ведущего AS-i устройства CM1243-2 и вызвать "Конфигурацию устройства". Затем нажать на кнопку "Выйти в онлайн" на панели инструментов. После в меню "Режим онлайн" выбрать команду "Онлайн и диагностика".



Существует два режима работы AS-i:

- Безопасный режим:
  - Нельзя изменить ведомое AS-i устройство и адреса ввода-вывода CPU.
  - Зеленый светодиод "CM" отключен.
- Режим конфигурирования:
  - Можно вносить требуемые изменения в свое ведомое AS-i устройство и адреса ввода-вывода CPU.
  - Зеленый светодиод "CM" включен.



В поле "Определение адреса AS-i" можно изменить адрес ведомого AS-i устройства. Новое ведомое устройство, которому еще не был назначен адрес, всегда имеет адрес 0. Оно обнаруживается ведущим устройством как новое ведомое устройство без назначения адреса и не включается в нормальную коммуникацию, пока не будет назначен адрес.

Ошибка конфигурирования

Если желтый светодиод "CER" светится, то это указывает на ошибку в конфигурации ведомого AS-i устройства. Нажать кнопку "ФАКТИЧЕСКАЯ > ЗАДАННАЯ", чтобы заменить конфигурацию ведомого устройства от модуля ведущего AS-i устройства CM1243-2 на конфигурацию ведомого устройства полевой шины AS-i.

11.5 Коммуникация S7

11.5.1 Инструкции GET и PUT (чтение и запись из удаленного CPU)

С помощью инструкций GET и PUT можно обмениваться данными с S7-CPU, используя соединения PROFINET и PROFIBUS. Это возможно только в том случае, если функция "Разрешить доступ с помощью связи PUT/GET" была активирована для партнерского CPU в разделе "Защита" свойств локального CPU:

- Обращение к данным в удаленном CPU: S7-1200 CPU может использовать в поле ввода ADDR_x только абсолютные адреса для обращения к переменным удаленных CPU (S7-200/300/400/1200).
- Обращение к данным в стандартном DB: S7-1200 CPU может использовать в поле ввода ADDR_x только абсолютные адреса для обращения к переменным DB в стандартном DB удаленного S7-CPU.
- Обращение к данным в оптимизированном DB: S7-1200 CPU не может обращаться к переменным DB в оптимизированном DB удаленного S7-1200 CPU.
- Обращение к данным в локальном CPU: S7-1200 CPU может использовать абсолютные или символьные адреса в качестве вводных данных в поле ввода RD_x или SD_x инструкции GET или PUT.

Примечание

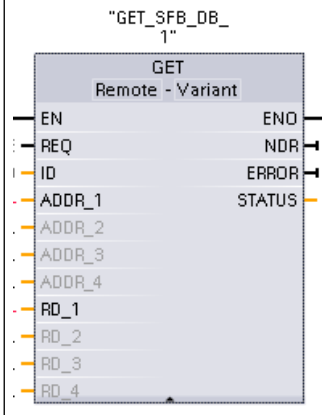
Функция GET/PUT не активируется автоматически в программе CPU версии V4.0

Функция GET/PUT программы CPU версии V3.0 автоматически активирована в CPU версии V4.0.

Но функция GET/PUT программы CPU версии V4.0 не активирована автоматически в CPU версии V4.0. Необходимо разрешить доступ GET/PUT (Страница 203) в конфигурации устройства CPU на вкладке "Свойства" в окне инспектора в пункте свойств "Защита".

Таблица 11-68 Инструкции GET и PUT

LAD/FBD	SCL	Описание

	`"GET_DB"(`
	`req:=_bool_in_,`
	`ID:=_word_in_,`
	`ndr=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`addr_1:=_remote_inout_,`
	`[...addr_4:=_remote_inout_,]`
	`rd_1:=_variant_inout_`
	`[,...rd_4:=_variant_inout_]);`
		С помощью инструкции GET считываются данные из удаленного S7-CPU. Удаленный CPU может находиться в режиме работы RUN или STOP.
		При вставке инструкции STEP 7 автоматически создает DB.

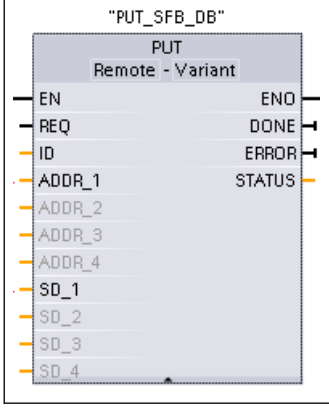
	`"PUT_DB"(`
	`req:=_bool_in_,`
	`ID:=_word_in_,`
	`done=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`addr_1:=_remote_inout_,`
	`[...addr_4:=_remote_inout_,]`
	`sd_1:=_variant_inout_,`
	`[....sd_4:=_variant_inout_]);`
		С помощью инструкции PUT записываются данные в удаленный S7-CPU. Удаленный CPU может находиться в режиме работы RUN или STOP.
		При вставке инструкции STEP 7 автоматически создает DB.

Таблица 11-69 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	Input	Bool	Инструкцию запускает положительный фронт (0 на 1).
ID	Input	CONN_PRG (Word)	ID S7 соединения (шестн.)
NDR (GET)	Output	Bool	Новые данные готовы: • 0: Запрос еще не запущен или еще выполняется. • 1: Задача была успешно завершена.
DONE (PUT)	Output	Bool	DONE: • 0: Запрос еще не запущен или еще выполняется. • 1: Задача была успешно завершена.

Параметр и тип данных		Тип данных	Описание
ERROR STATUS	Output Output	Bool Word	• ERROR=0 Значение STATUS: – 0000H: Предупреждения и ошибки отсутствуют. – <> 0000H: Предупреждение, подробная информация в STATUS. • ERROR=1 Ошибка. Предупреждение, подробная информация о природе ошибки в STATUS.
ADDR_1	InOut	Remote	Указатель на области памяти в удаленном CPU, которые хранят данные для чтения (GET) или передачи (PUT).
ADDR_2	InOut	Remote	
ADDR_3	InOut	Remote	
ADDR_4	InOut	Remote	
RD_1 (GET) SD_1 (PUT)	InOut	Variant	Указатель на области памяти в локальном CPU, которые хранят данные для чтения (GET) или передачи (PUT). Допустимые типы данных: Bool (разрешен только одиночный бит), Byte, Char, Word, Int, DWord, DInt или Real. Примечание: Если указатель обращается к блоку данных, то всегда должен указываться абсолютный адрес, например: P# DB10.DBX5.0 Byte 10 В этом случае значение 10 - это количество байт для инструкции GET или PUT.
RD_2 (GET) SD_2 (PUT)	InOut	Variant	
RD_3 (GET) SD_3 (PUT)	InOut	Variant	
RD_4 (GET) SD_4 (PUT)	InOut	Variant	

Необходимо обеспечить соответствие по длине (число байт) и типу данных между параметрами ADDR_x (удаленный CPU) и RD_x или SD_x (локальный CPU). Число после идентификатора "Byte" - это число байт, указываемое параметром ADDR_x, RD_x или SD_x .

Примечание

Общее количество байт, полученных инструкцией GET, или общее количество байт, отправленных инструкцией PUT, ограничено. Ограничение зависит от того, сколько из четырех возможных диапазонов адресов и областей памяти используются:

- Если используются только ADDR_1 и RD_1/SD_1, то инструкция GET может получить 222 байта, а инструкция PUT может передать 212 байт.
- Если используются ADDR_1, RD_1/SD_1, ADDR_2 и RD_2/SD_2, то инструкция GET может получить в общей сложности 218 байт, а инструкция PUT может передать в общей сложности 196 байт.
- Если используются ADDR_1, RD_1/SD_1, ADDR_2, RD_2/SD_2, ADDR_3 и RD_3/SD_3 , то инструкция GET может получить в общей сложности 214 байтов, а инструкция PUT может передать в общей сложности 180 байтов.
- Если используются ADDR_1, RD_1/SD_1, ADDR_2, RD_2/SD_2, ADDR_3, RD_3/SD_3, ADDR_4, RD_4/SD_4, то инструкция GET может получить в общей сложности 210 байт, а инструкция PUT может передать в общей сложности 164 байта.

Сумма числа байтов каждого из параметров диапазона адресов и области памяти должна быть меньше или равна установленным предельным значениям. При превышении этих предельных значений, инструкция GET или PUT возвращает ошибку.

При положительном фронте в параметре REQ инструкция чтения (GET) или инструкция записи (PUT) загружает параметры ID, ADDR_1 и RD_1 (GET) или SD_1 (PUT).

- Для GET: Удаленный CPU передает запрошенные данные в области приема (RD_x). Тем самым процесс начинается в следующем цикле. Если инструкция чтения завершилась с ошибкой, то параметр NDR устанавливается на "1". Новая инструкция может быть запущена только после завершения предыдущей инструкции.
- Для PUT: Локальный CPU начинает передавать данные (SD_x) на адрес (ADDR_x) в памяти удаленного CPU. После успешного завершения инструкции записи, удаленный CPU возвращает подтверждение выполнения. Параметр DONE инструкции PUT после устанавливается на "1". Новая инструкция записи может быть запущена только после завершения предыдущей инструкции.

Примечание

Для обеспечения целостности данных всегда следует проверять, была ли инструкция завершена (NDR = 1 для GET или DONE = 1 для PUT), прежде чем обращаться к данным или выполнять другую инструкцию чтения или записи.

Параметры ERROR и STATUS предоставляют информацию о состоянии инструкции чтения (GET) или записи (PUT).

Таблица 11-70 Информация об ошибке

ERROR	STATUS (дес.)	Описание
0	11	• Новая операция не может быть активирована, так как предшествующая операция еще не завершена. • Операция уже обрабатывается в более низком классе приоритета.
0	25	Коммуникация запущена. Операция находится в обработке.
1	1	Проблемы со связью, например: • Не загружено описание соединения (локального или удаленного) • Соединение прервано (пример: кабель, CPU выключен или CM/CB/CP в состоянии STOP) • Соединение с участником процесса коммуникации еще не установлено
1	2	Отрицательное квитирование от устройства-партнера. Задание не может быть выполнено
1	4	Ошибка в указателях области передачи (RD_x для GET, SD_x для PUT) в части длины или типа данных.
1	8	Ошибка доступа на партнерском CPU
1	10	Доступ к локальной памяти пользователя невозможен (пример: попытка доступа к отсутствующему блоку данных)
1	12	После вызова SFB: • Указан блок данных экземпляра, который не относится к GET или PUT. • Указан не DB экземпляра, а DB общего доступа • Не найден блок данных экземпляра (загрузка нового DB экземпляра)

ERROR	STATUS (дес.)	Описание
1	20	• Превышено максимальное количество параллельных операций/экземпляров • Экземпляры были повторно загружены при CPU-RUN Это состояние возможно при первом выполнении инструкции GET или PUT.
1	27	Отсутствует соответствующая инструкция GET или PUT в CPU.

11.5.2 Создание соединения S7

Механизмы соединения

Чтобы получить доступ к удаленным партнерам по соединению с помощью инструкции PUT/GET, пользователь должен обладать соответствующими разрешениями.

По умолчанию опция "Разрешить доступ с использованием коммуникации PUT/GET" отключена. В этом случае доступ по чтению и записи к данным CPU возможен только для коммуникационных соединений, которые требуют конфигурации или программирования как для локального CPU, так и для другого участника процесса коммуникации (партнера). Например, возможен доступ к помощью инструкций BSEND/BRCV.

Соединения, для которых локальный CPU является только сервером (подразумевается, что никакого сконфигурированного/запрограммированного соединения с коммуникационным партнером не существует в локальном CPU) во время работы CPU невозможны. Это относится, напр., к:

- PUT/GET-, FETCH/WRITE- или FTP-доступу через коммуникационные модули
- PUT/GET-доступу из других S7-CPU
- доступу через HMI с использование PUT/GET-коммуникации

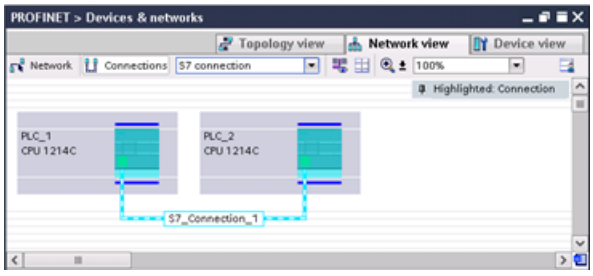
Если необходим доступ к данным CPU со стороны клиента, т.е. коммуникационные возможности CPU не должны ограничиваться, можно сконфигурировать защиту от несанкционированного доступа для S7-1200 CPU (Страница 203) для этого уровня безопасности.

Типы соединений

Выбранный тип соединения создает коммуникационное соединение со станцией-партнером. Соединение настраивается, устанавливается и автоматически контролируется.

На портале "Устройства и сети" в "Просмотре сетевых соединений" можно объединять в сеть устройства в проекте. Сначала следует открыть вкладку "Соединения" и выбрать в выпадающем списке справа тип соединения (например, S7-соединение). Кликнуть по зеленому полю (PROFINET) на первом устройстве и провести линию к полю PROFINET на втором устройстве. Отпустить кнопку мыши. Соединение PROFINET создано.

Дополнительную информацию см. "Создание сетевого соединения" (Страница 811).



Нажать на кнопку "Выделено: Соединение" для вызова диалогового окна конфигурирования "Свойства" коммуникационной инструкции.

11.5.3 Конфигурирование пути соединения между локальным и партнерским CPU

Конфигурирование общих параметров

Параметры коммуникации определяются в диалоговом окне "Свойства" коммуникационной инструкции. Это диалоговое окно отображается в нижней части страницы каждый раз, когда выбирается любая часть инструкции.

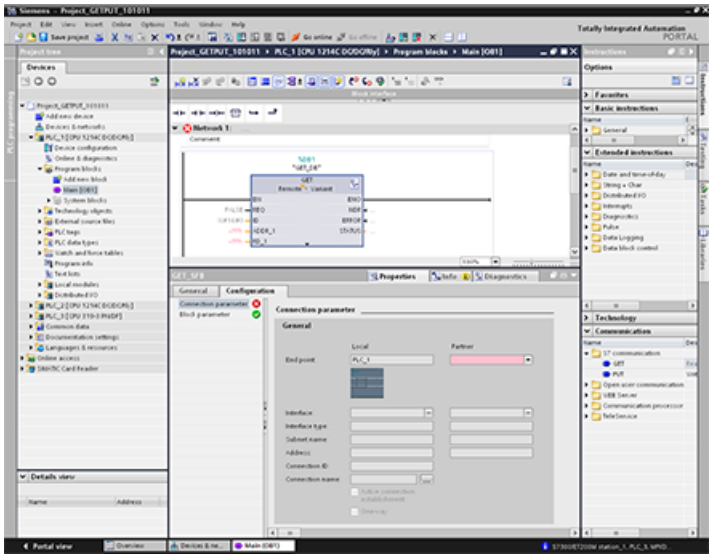
Дополнительную информацию см. в "Конфигурация устройства: конфигурирование пути соединения между локальным и партнерским CPU (Страница 812)".

Используемые TSAP или порты определяются в диалоговом окне "Параметры соединения" в пункте "Адресная информация". TSAP или порт соединения в CPU вводится в поле "Локальный TSAP". TSAP или порт для соединения в партнерском CPU вводится в поле "Партнерская TSAP".

11.5.4 Назначение параметров соединения для GET/PUT

Назначение параметров соединения для инструкций GET/PUT помогает пользователю при конфигурировании S7-соединений для S7 коммуникации CPU-CPU.

STEP 7 после вставки блока GET или PUT отображает диалоговое окно назначения для параметров соединения инструкций GET/PUT:



Окно инспектора выводит на экран свойства соединения каждый раз при выборе части инструкции. Параметры коммуникации определяются на вкладке "Конфигурация" в диалоге "Свойства" коммуникационной инструкции.

Примечание

Функция GET/PUT не активирована по умолчанию в программе CPU версии V4.1 и выше

Функция GET/PUT в программе CPU версии V3.0 автоматически активирована в CPU версии V4.1 и выше.

Но функция GET/PUT в программе CPU версии V4.1 не активирована автоматически в CPU версии V4.1 и выше. Необходимо разрешить доступ GET/PUT (Страница 203) в конфигурации устройства CPU на вкладке "Свойства" в окне инспектора в пункте свойств "Защита".

11.5.4.1 Параметры соединения

На странице "Параметры соединения" можно сконфигурировать требуемое S7-соединение и параметр "Идентификатор соединения", на который ссылается параметр "ID" блока GET/PUT. Страницы содержит информацию о локальной конечной точке и позволяет определить локальный интерфейс. Также можно определить конечную точку участника.

На странице "Параметры блока" можно сконфигурировать дополнительные параметры блока.

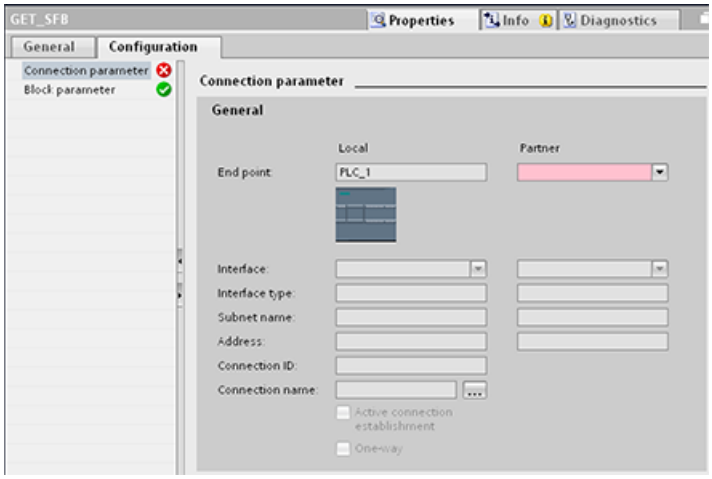


Таблица 11-71 Параметры соединения: Общие определения

Параметр		Определение
Параметры соединения: Общее	Конечная точка	"Локальная конечная точка": Имя локального CPU "Удаленная конечная точка": Имя партнерского CPU (удаленного) Примечание: В выпадающем списке "Удаленная конечная точка" система отображает всех потенциальных участников S7-соединения текущего проекта, а также опцию "Не указан". Неуказанный участник представляет собой коммуникационного партнера, который отсутствует в настоящее время в проекте STEP 7 (например, коммуникационный партнер в виде стороннего устройства).
	Интерфейс	Название интерфейсов Примечание: Можно изменить соединение, изменив локальный и удаленный интерфейсы
	Тип интерфейса	Тип интерфейса
	Имя подсети	Имя подсетей
	Адрес	Назначенные IP-адреса Примечание: Для "не указанного" участника процесса коммуникации может быть указан удаленный адрес стороннего устройства.
	Идентификатор соединения	Номер ID: Автоматически генерируется при назначении параметров соединения GET/PUT
	Имя соединения	Адрес памяти данных для локального CPU и партнерского CPU: Автоматически генерируется при назначении параметров соединения GET/PUT
	Активное установление соединения	Флажок для выбора локального CPU в качестве активного соединения
	Одностороннее	Флажок для определения одностороннего или двухстороннего соединения Примечание: В соединении GET/PUT с PROFINET, как локальное, так и партнерское устройства могут действовать как сервер или клиент. Это разрешает двухстороннее соединение, и флажок "Одностороннее" снят. В соединении GET/PUT с PROFIBUS, в некоторых случаях, партнерское устройство может работать только как сервер (например, в случае S7-300), и флажок "Одностороннее " установлен.

Параметр "ID соединения"

Есть три способа изменить определенные системой ID соединения:

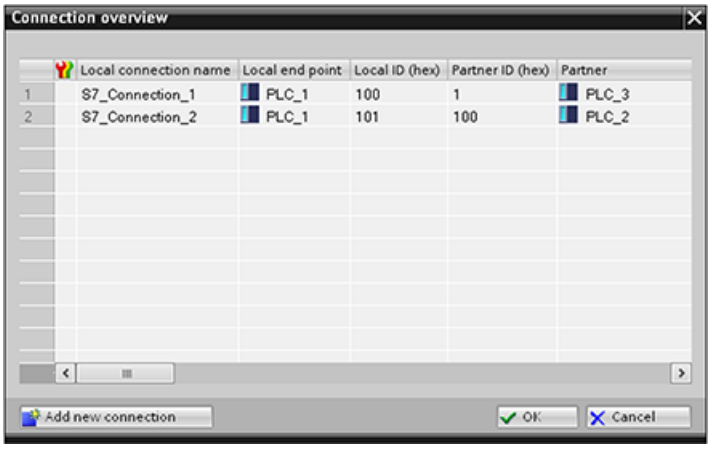
1. Пользователь может изменить текущий ID непосредственно в блоке GET/PUT. Если новый ID относится к уже существующему соединению, соединение изменяется.
2. Пользователь может изменить текущий ID непосредственно на блоке GET/PUT, но новый ID еще не существует. Новое S7-соединение создается системой.
3. Пользователь может изменить текущий ID через диалоговое окно "Обзор соединений": вводимые пользователем данные синхронизируются с ID-параметром на соответствующем блоке GET/PUT.

Примечание

Параметр "ID" блока GET/PUT не является именем соединения, а числовым выражением, которое записывается как в следующем примере: W#16#1

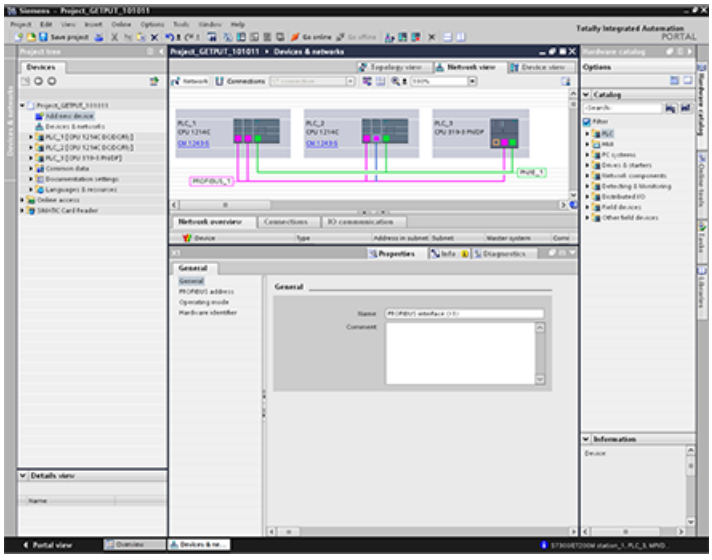
Параметр "Имя соединения"

Имя соединения может редактироваться с помощью специального элемента управления, диалогового окна "Обзор соединений". Это диалоговое окно предлагает все доступные S7-соединения, которые могли бы быть выбраны как альтернатива для текущей GET/PUT коммуникации. Пользователь может создать абсолютно новое соединение в этой таблице. Нажать на кнопку справа от поля "Имя соединения", чтобы открыть диалоговое окно "Обзор соединений".



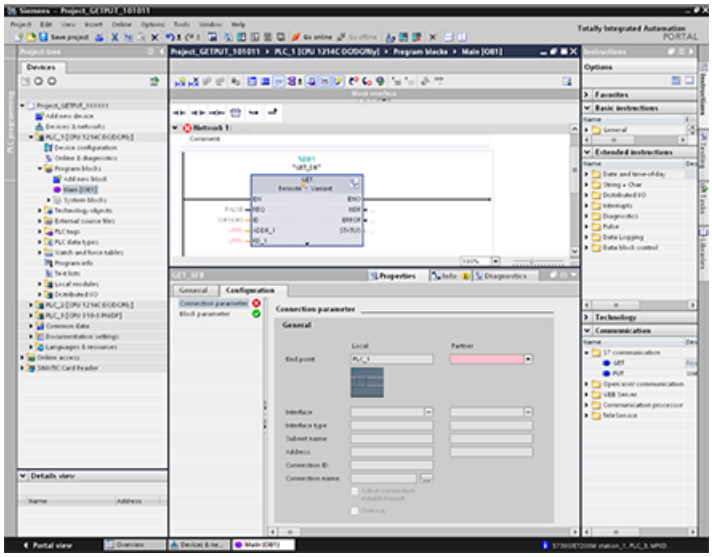
11.5.4.2 Конфигурирование S7-соединения между CPU

Учитывая конфигурацию PLC_1, PLC_2 и PLC_3, как показано на рисунке ниже, вставить блоки GET или PUT для "PLC_1".



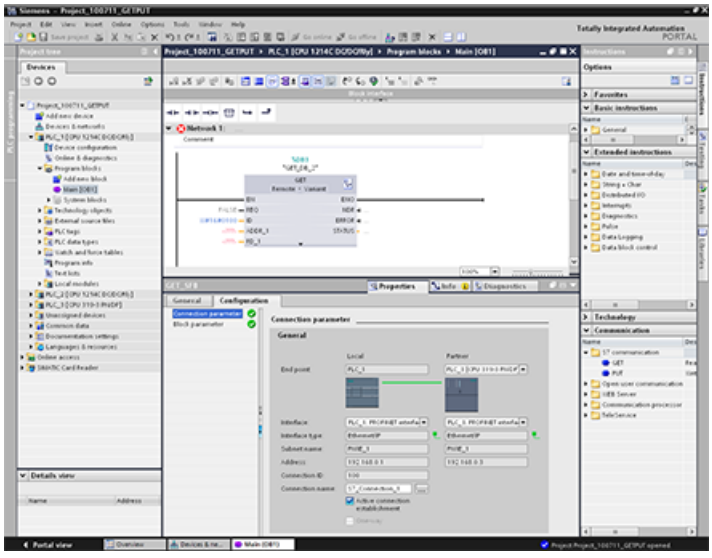
Для инструкции GET или PUT в окне инспектора автоматически отображается вкладка "Свойства" со следующими элементами меню:

- "Конфигурация"
- "Параметры соединения"



Конфигурирование S7-соединение PROFINET

Выбрать для "Удаленной конечной точки" опцию "PLC_3".



Система реагирует со следующими изменениями:

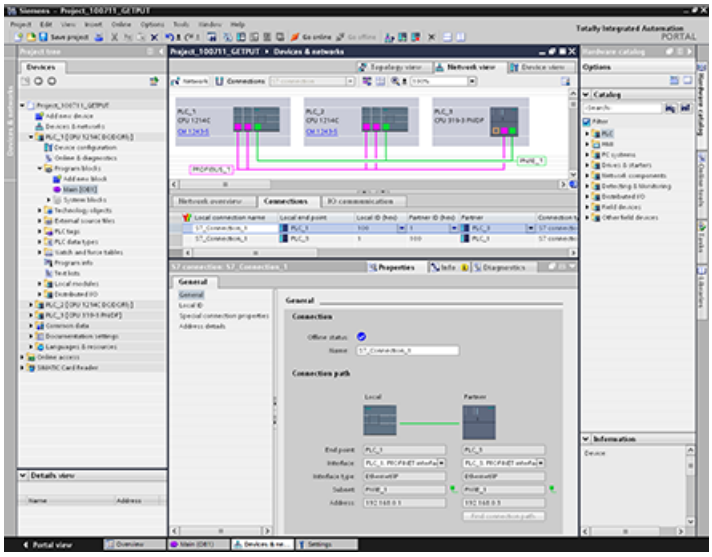
Таблица 11-72 Параметры соединения: Общие значения

Параметр		**Определение**
Параметры соединения: Общее	Конечная точка	"Локальная конечная точка" содержит "PLC_1" (с защитой от записи). Поле "Удаленная конечная точка" содержит "PLC_3[CPU319-3PN/DP]": • Цвет изменяется с красного на белый. • Появляется изображение "партнерского" устройства. • Соединительная линия появляется между изображениями устройств PLC_1 и PLC_3 (зеленая линия Ethernet).
	Интерфейс	"Локальный интерфейс" содержит "CPU1214C DC/DC/DC, интерфейс PROFINET (R0/S1)". "Удаленный интерфейс" содержит: "CPU319-3PN/DP, PROFINET интерфейс (R0/S2)".
	Тип интерфейса	"Локальный тип интерфейса" содержит "Ethernet/IP"; управление с защитой от записи. "Удаленный тип интерфейса" содержит "Ethernet/IP"; управление с защитой от записи. Изображения типа интерфейса показаны справа от локального и удаленного типа интерфейса (зеленый значок Ethernet).
	Имя подсети	"Имя локальной подсети" содержит "PN/IE_1"; управление с защитой от записи. "Имя удаленной подсети" содержит "PN/IE_1"; управление с защитой от записи.
	Адрес	"Локальный адрес" содержит локальный IP-адрес; управление с защитой от записи. "Удаленный адрес" содержит удаленный IP-адрес; управление с защитой от записи.
	Идентификатор соединения	"Идентификатор соединения" содержит "100". В главной программе [OB1] редактора текстов программы "Идентификатор соединения" блока GET/PUT также содержит значение "100".
	Имя соединения	"Имя соединения" содержит имя соединения по умолчанию (пример: "S7_Connection_1"); управление включено.
	Активное установление соединения	Флажок для выбора локального CPU в качестве активного соединения.
	Одностороннее	С защитой от записи и не включено. Примечание: "PLC_1" (S7-1200 CPU 1214CDC/DC/реле) и "PLC_3" (S7-300 CPU 319-3PN/DP) в PROFINET соединении GET/PUT оба могут действовать и как сервер, и как клиент, что делает возможным двустороннее соединение.

Значок GET/PUT в дереве просмотра свойств также меняет цвет с красного на зеленый.

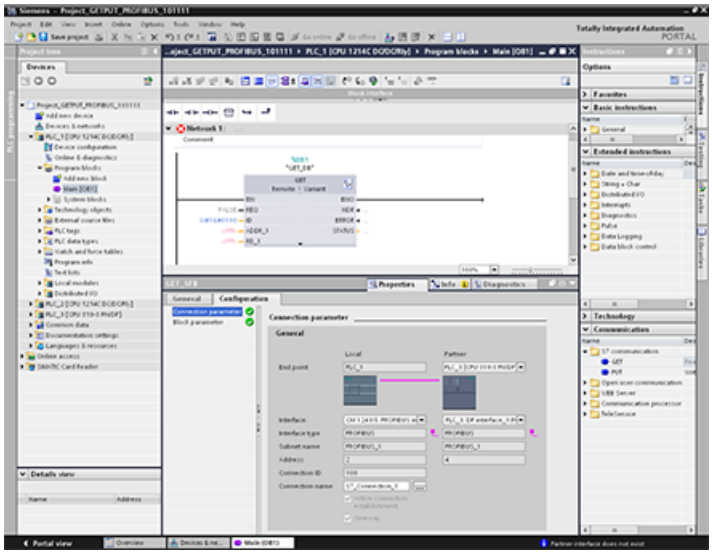
Завершенное S7-соединение PROFINET

В "Просмотре сетевых соединений" в таблице "Соединения" между "PLC_1" и "PLC_3" отображается двухстороннее S7-соединение.



Конфигурирование S7-соединения PROFIBUS

Выбрать для "Удаленной конечной точки" опцию "PLC_3".



Система реагирует со следующими изменениями:

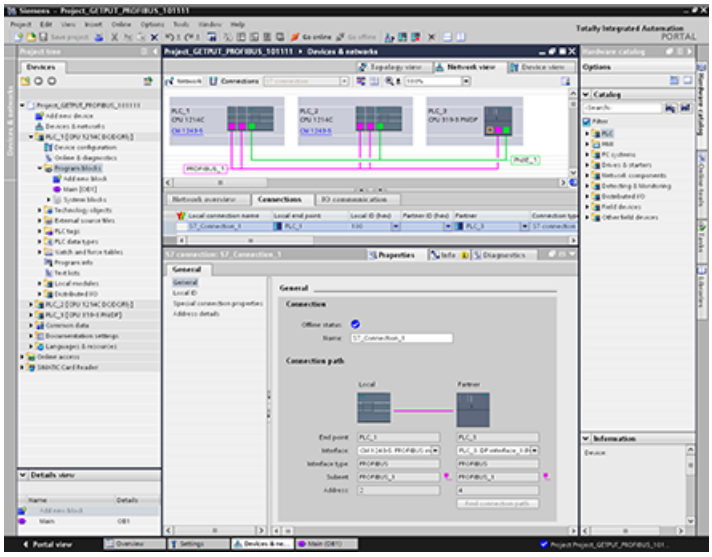
Таблица 11-73 Параметры соединения: Общие значения

Параметр		**Определение**
Параметры соединения: Общее	Конечная точка	"Локальная конечная точка" содержит "PLC_1" (с защитой от записи). Поле "Удаленная конечная точка" содержит "PLC_3[CPU319-3PN/DP]": • Цвет изменяется с красного на белый. • Появляется изображение "партнерского" устройства. • Соединительная линия появляется между изображениями устройств PLC_1 и PLC_3 (фиолетовая линия PROFIBUS).
	Интерфейс	"Локальный интерфейс" содержит "CPU1214C DC/DC/DC, интерфейс PROFIBUS (R0/S1)". "Удаленный интерфейс" содержит: "CPU319-3PN/DP, PROFIBUS интерфейс (R0/S2)".
	Тип интерфейса	"Локальный тип интерфейса" содержит "PROFIBUS"; управление с защитой от записи. "Удаленный тип интерфейса" содержит "PROFIBUS"; управление с защитой от записи. Изображения типа интерфейса показаны справа от локального и удаленного типа интерфейса (фиолетовый значок PROFIBUS).
	Имя подсети	"Имя локальной подсети" содержит "PROFIBUS_1"; управление с защитой от записи. "Имя удаленной подсети" содержит "PROFIBUS_1"; управление с защитой от записи.
	Адрес	"Локальный адрес" содержит локальный IP-адрес; управление с защитой от записи. "Удаленный адрес" содержит удаленный IP-адрес; управление с защитой от записи.
	Идентификатор соединения	"Идентификатор соединения" содержит "100". В главной программе [OB1] редактора текстов программы "Идентификатор соединения" блока GET/PUT также содержит значение "100".
	Имя соединения	"Имя соединения" содержит имя соединения по умолчанию (пример: "S7_Connection_1"); управление включено.
	Активное установление соединения	С защитой от записи и включено для выбора локального CPU в качестве активного соединения.
	Одностороннее	С защитой от записи и включено. Примечание: "PLC_3" (S7-300 CPU319-3PN/DP) в соединении GET/PUT с PROFIBUS может быть только сервером (не может быть клиентом), поэтому возможно только одно одностороннее соединение.

Значок GET/PUT в дереве просмотра свойств также меняет цвет с красного на зеленый.

Завершенное PROFIBUS S7-соединение

В "Просмотре сетевых соединений" в таблице "Соединения" между "PLC_1" и "PLC_3" отображается одностороннее S7-соединение.



11.6 Что делать, если доступ к CPU по IP-адресу невозможен

Если CPU недоступен по IP-адресу, можно настроить временный аварийный IP-адрес для CPU. Через аварийный IP-адрес можно восстановить соединение с CPU для загрузки конфигурации устройства с правильным IP-адресом.

Причины, когда может потребоваться аварийный IP-адрес

CPU может быть недоступен после загрузки проекта с одной из следующих проблем:

- IP-адрес PROFINET интерфейса CPU совпадает с адресом другого устройства в сети.
- Неправильная подсеть для CPU.
- Маска подсети делает CPU недоступным.

В этих случаях CPU становится недоступным для STEP 7.

Назначение аварийного IP-адреса

Аварийный IP-адрес может быть назначен в следующих случаях:

- В конфигурации устройства в STEP 7 для IP-протокола выбрано "Устанавливать IP-адрес в проекте".
- Модуль CPU находится в рабочем состоянии STOP.

В этих случаях для IP-адреса устройства с помощью инструмента DCP может быть установлен аварийный IP-адрес. Например, в SIMATIC Automation Tool есть DCP команда "Set IP address". Аварийный IP-адрес может быть установлен независимо от степени защиты (Страница 203) CPU. После установки временного IP-адреса с помощью инструмента DCP, на CPU включается светодиод обслуживания. В буфер диагностики записывается, что для интерфейса Ethernet активирован аварийный адрес.

Восстановление IP-адреса после назначения аварийного IP-адреса

Буфер диагностики предоставляет информацию о включении или выключении аварийного IP-адреса. Для сброса аварийного IP-адреса также можно выключить и включить CPU.

После присвоения аварийного IP-адреса можно загрузить проект STEP 7 с правильным IP-адресом для CPU. После загрузки проекта следует выключить и снова включить CPU.

11.7 Сервер OPC UA

OPC UA-коммуникация поддерживается прошивкой S7-1200 CPU от версии 4.4. CPU обеспечивает доступ к данным путем поддержки конфигурации в качестве сервера OPC UA.

OPC Foundation (https://opcfoundation.org/) предоставляет спецификации для основных функций OPC UA-коммуникации. S7-1200 CPU поддерживают часть функций OPC UA. Ниже будет описана эта часть функций OPC UA.

См. также

Новые функции (Страница 35)

11.7.1 Конфигурирование сервера OPC UA

Сервер OPC UA может быть сконфигурирован в TIA Portal в свойствах оборудования CPU.

11.7.1.1 Активация сервера OPC UA

Сервер OPC UA по умолчанию не активирован. Необходимо явно активировать сервер OPC UA путем установки флажка "Активировать сервер OPC UA" в свойствах оборудования CPU.

11.7.1.2 Настройки для сервера OPC UA

В следующей таблице перечислены настройки для конфигурирования сервера OPC UA. Кроме этого, таблица содержит ограничения сервера для S7-1200 с прошивкой версии V4.4:

Настройки сервера	Мин.	Макс.	Настройка по умолчанию	Настраивается пользователем?
Порты передачи данных	1024	49151	4840	Да
Макс. тайм-аут сеанса	1 с	600.000 с	30 с	Да
Одновременные сеансы	1	5	2	Да
Узлы на интерфейсе сервера[1]		1000		Нет
Макс. число зарегистрированных узлов[2]	100	1000	200	Да
Общее число пространств имен[3]		18		Нет
Ограничения подписки				
Мин. интервал считывания	100 мс, 250 мс, 500 мс, 1000 мс, 5000 мс, 10000 мс		1000 мс	Да
Мин. интервал публикации	200 мс	20.000.000 мс	1000 мс	Да
Одновременные подписки		5		Нет
Макс. число контролируемых узлов	100	500	200	Да

[1] Это максимальное число определенных пользователем узлов. Сюда не включены узлы, не явно определенные на интерфейсе сервера.

[2] "Зарегистрированные" узлы конфигурируются для быстрого просмотра. Сервер резервирует место в памяти для таблицы зарегистрированных узлов. Размер этой резервируемой области зависит от установки пользователя "Макс. число зарегистрированных узлов".

[3] Это общее количество определенных пространств имен для всех интерфейсов сервера и ссылочных пространств имен.

11.7.2 Безопасность сервера OPC UA

Для активации сервера OPC UA потребуется сертификат. При активации сервера, TIA Portal генерирует сертификат автоматически. Этот сертификат может быть изменен в свойствах PLC.

11.7.2.1 Поддерживаемые правила безопасности

Можно выбрать, какие правила безопасности будут поддерживаться сервером. Правила безопасности, которые выбирает клиент OPC UA во время выполнения, определяют защиту связи между клиентом и сервером.

S7-1200 поддерживает следующие правила безопасности:

Правило безопасности	Включено по умолчанию?
Защита отсутствует	Да
Basic128Rsa15 – Sign	Нет
Basic128Rsa15 – Sign & Encrypt	Нет

Правило безопасности	Включено по умолчанию?
Basic256 – Sign	Нет
Basic256 – Sign & Encrypt	Нет
Basic256Sha256 – Sign	Да
Basic256Sha256 – Sign & Encrypt	Да

11.7.2.2 Доверенные клиенты

По умолчанию сервер OPC UA разрешает подключения для любых клиентов.

В качестве альтернативы можно определить список доверенных клиентов. Доверенные клиенты определяются по их сертификатам. После выбора при работе только клиенты с доверенными сертификатами смогут подключаться к серверу.

Для того, чтобы указать доверенных клиентов, следует добавить их сертификаты в список "Доверенные клиенты" в TIA Portal в пункте "Свойства оборудования > Безопасность OPC UA > Защищенный канал > Доверенные клиенты".

11.7.2.3 Аутентификация пользователя

S7-1200 поддерживает как гостевую аутентификацию, так и аутентификацию по имени и паролю пользователя. По умолчанию используется гостевая аутентификация.

Когда гостевая аутентификация включена, клиенту не требуется указывать имя пользователя и пароль во время подключения.

Когда гостевая аутентификация отключена, разрешено подключение только клиентам, которые предоставляют ранее указанные имя пользователя и пароль.

Действительные пользователи для сервера OPC UA могут определяться двумя способами:

- Действительные пользователи определяются в свойствах оборудования CPU.
- Если была включена глобальную защиту для проекта, то действительные пользователи будут определены по ролям, которые были указаны для отдельных пользователей проектов TIA Portal. Можно сконфигурировать до 21 пользователя.

11.7.3 Интерфейс сервера OPC UA

Сервер OPC UA S7-1200 не поддерживает стандартный интерфейс сервера SIMATIC. Вместо этого необходимо определить структуру и содержимое интерфейса сервера в TIA Portal, а затем загрузить его в PLC.

Для добавления интерфейса сервера действовать следующим образом:

- Кликнуть в дереве проекта по имени PLC.
- Выбрать "OPC UA-коммуникация".
- Выбрать "Интерфейсы сервера".
- Выбрать "Добавление нового интерфейса сервера".

- Ввести необходимую для сервера информацию.
- Загрузить информацию сервера в PLC.

11.7.3.1 Поддерживаемые типы данных

OPC Foundation (https://opcfoundation.org/) определяет набор поддерживаемых типов данных, которые описывают структуру атрибута Value переменных и их типов. S7-1200 версии V4.4 поддерживает подмножество этих типов данных, а также другие определенные типы, которые являются производными от этих типов данных.

В таблице ниже перечислены поддерживаемые S7-1200 версии V4.4 типы данных:

Тип SIMATIC	Имя типа OPC UA	Идентификатор узла
Bool	Булево значение	i=1
SInt	SByte	i=2
USInt	Byte	i=3
Int	Int16	i=4
UInt	UInt16	i=5
DInt	Int32	i=6
UDInt	UInt32	i=7
Real	Float	i=10
LReal	Double	i=11
WString	String	i=12
DWord	StatusCode	i=19

Следует учитывать, что этот список содержит поддерживаемые базовые типы узлов. Он не является полным списком поддерживаемых типов узлов, поскольку многие типы данных SIMATIC могут быть назначены базовым типам узлов. Любой тип данных SIMATIC, который может быть назначен базовому типу узла, также является поддерживаемым типом узла.

Интерфейс сервера S7-1200 поддерживает чтение и запись узлов с простыми типами данных. Следующие типы не поддерживаются:

- Структурированные типы данных
- Массивы
- Объединения
- Методы сервера

S7-1200 принимает загрузку сервера с неподдерживаемыми типами. Однако PLC возвращает ошибку, если клиент пытается прочитать или записать на узел с неподдерживаемым типом.

11.7.3.2 Представление PLC

Интерфейс сервера OPC UA предоставляет узлы, представляющие PLC, со свойствами, описывающими PLC. Они доступны при активации сервера OPC UA.

Следующая информация предоставляется интерфейсом сервера:

Атрибут	**Значение**		
NodeId	Ns=SI;s=PLC		
BrowseName	SI:<PLC> Где <PLC> - это имя, присвоенное PLC пользователем в проекте TIA Portal.		
References	**NodeClass**	**BrowseName**	**Kommentar**
ComponentOf the DeviceSet Object			
OrganizedBy the Objects folder			
HasTypeDefinition	ObjectType	SimaticDeviceType	Производное от типа устройства
HasProperty	Variable	DeviceManual	
HasProperty	Variable	DeviceRevision	
HasProperty	Variable	EngineeringRevision	
HasProperty	Variable	HardwareRevision	
HasProperty	Variable	Symbol	
HasProperty	Variable	Manufacturer	
HasProperty	Variable	Model	
HasProperty	Variable	OperatingMode	
HasProperty	Variable	OrderNumber	
HasProperty	Variable	RevisionCounter	
HasProperty	Variable	SerialNumber	
HasProperty	Variable	SoftwareRevision	

11.7.3.3 Загружаемые интерфейсы сервера

Компоненты интерфейса сервера OPC UA создаются и редактируются в TIA Portal. Интерфейсы сервера OPC UA могут определяться двумя способами:

- Интерфейсы могут быть основаны на созданных на внешних устройствах XML-файлах (известные как спецификации компаньона)
- Интерфейсы могут быть определены непосредственно в TIA Portal на основе элементов блока данных и глобальных переменных (тегов), включенных в программу.

При загрузке в целевую систему таким образом определяется интерфейс по данным, видимый клиенту OPC UA.

Требуется соглашение об использовании

Для запуска сервера OPC UA на S7-1200 CPU потребуется соглашение об использовании. Доступны следующие соглашения (лицензии):

- SIMATIC OPC UA S7-1200 Basic DVD 6ES7823-0BA00-2BA0
- SIMATIC OPC UA S7-1200 Basic DL 6ES7823-0BE00-2BA0

Требуемый тип лицензии отображается в "Свойства> Общее > Соглашения об использовании > OPC-UA > Тип требуемой лицензии". Для того, чтобы подтвердить приобретение требуемой лицензии, выполнить следующие действия:

1. Кликнуть в свойствах CPU по "Соглашения об использовании > OPC UA".
2. Выбрать из раскрывающегося списка "Тип приобретенной лицензии" требуемую лицензию.

Веб-сервер

12

Веб-сервер S7-1200 обеспечивает веб-доступ к данным CPU и данным процесса.

Веб-страницы S7-1200 могут вызываться из PC или через мобильное устройство. Для устройств с маленьким дисплеем у веб-сервера есть коллекция базовых страниц (Страница 1037).



Для обращения к IP-адресу S7-1200 CPU или IP-адресу активированного через веб-сервер CP модуля (коммуникационный процессор) (Страница 1033) в локальной стойке CPU для установления соединения следует использовать интернет-браузер. S7-1200 поддерживает несколько одновременных соединений.

Примечание

Несколько одновременных соединений с веб-сервером

Веб-сервер S7-1200 допускает 30 одновременных соединений. Экземпляры открытого браузера могут использовать от 2 до 8 соединений каждый. Веб-сервер поддерживает до 7 зарегистрированных пользователей, но Siemens рекомендует максимально сократить число пользователей, одновременно работающих с веб-сервером. При средней нагрузке 7 одновременных пользователей является приемлемым выбором.

Стандартные веб-страницы

S7-1200 содержит стандартные веб-страницы (Страница 1036), к которым можно обращатьсячерез интернет-браузер на PC (Страница 1029) или через мобильное устройство (Страница 1031):

- Вводная страница (Страница 1041) - Точка входа на стандартные веб-страницы
- Начальная страница (Страница 1042) - Общая информация о CPU
- Диагностика (Страница 1043) - Подробная информация о ЦПУ, включая серийный, заказной номера и номер версии, защиту программы и использование памяти
- Буфер диагностики
- Информация о модулях (Страница 1046) - Информация о локальных и удаленных модулях и возможности для обновления прошивки для локальных модулей
- Коммуникации (Страница 1050) - Информация о сетевых адресах, физических свойствах коммуникационных интерфейсов, статистика, параметры, а также обзор соединений и диагностическая информация

- Состояние переменных (тегов) (Страница 1055) - Переменные CPU и ввод-вывод, доступный по адресу или по имени переменной (тега) PLC
- Таблицы текущего контроля (Страница 1056) - Таблицы текущего контроля, сконфигурированные в STEP 7
- Резервное онлайн-копирование (Страница 1059) - Возможность для резервного копирования Онлайн-CPU или для восстановления имеющейся резервной копии
- Архивы данных (Страница 1061)- Возможность для просмотра списка всех архивов данных в PLC, загрузки архива данных из PLC на компьютер, удаления архива данных из PLC, а также вызова и удаления архива данных из PLC
- Файлы пользователя (Страница 1064)- Возможность для просмотра списка пользовательских файлов в PLC, загрузки пользовательского файла из PLC на свой компьютер, выгрузки пользовательского файла из компьютера на PLC и удаления пользовательского файла на PLC
- Определенные пользователем страницы (Страница 1070)- Создание пользовательских веб-страниц для доступа к данным CPU
- Браузер файлов (Страница 1068) - Браузер для файлов, хранящихся в CPU или на карте памяти, например, архивов данных и рецептов
- Вход в систему (Страница 1038) - Вход в систему под другим именем или выход из системы.

Эти страницы встроены в S7-1200 CPU и доступны на английском, немецком, французском, испанском, итальянском и упрощенном китайском языках. Для просмотра всех страницы, за исключением вводной и начальной страницы, потребуются дополнительныеправа пользователя (Страница 1028), конфигурируемые в STEP 7.

Примечание

Активация дополнительных языков для веб-сервера

Для отображения стандартных веб-страниц можно выбрать другие языки. Выбрать "Языки интерфейса пользователя" на вкладке "Свойства" конфигурации устройства, чтобы назначить язык проекта STEP 7 одному из шести поддерживаемых языков. После загрузки конфигурации устройства стандартные веб-страницы предлагают возможность для выбора языка интерфейса пользователя.

Пользовательские веб-страницы

Для S7-1200 можно создавать пользовательские веб-страницы для обращения к данным CPU. Можно разрабатывать эти страницы с помощью программного обеспечения для разработки HTML по выбору и включать предварительно определенные команды "AWP" (Automation Web Programming) в свой HTML-код для доступа к данным CPU. В главе Пользовательские веб-страницы (Страница 1070) можно найти специальную информацию о разработке пользовательских веб-страниц, а также о конфигурировании и программировании в STEP 7.

Для доступа к пользовательским страницам через вызов стандартных или базовых веб-страниц можно использовать PC или мобильное устройство. Также можно настроить одну из пользовательских веб-страниц как начальную страницу (Страница 1089) для веб-сервера.

Требования к интернет-браузеру

Siemens протестировал стандартные веб-страницы интернет-браузера и верифицировал поддержу следующих браузеров:

- Internet Explorer версии от 8 до 11
- Microsoft Edge
- Mozilla Firefox версий от V22 до V32, от V42 до V47
- Google Chrome версий от V33 до V38, от V46 до V47
- Mobile Safari и Mobile Chrome для устройств iOS 9
- Браузеры для Android следующих версий:
  - Jellybean V4.3
  - Kitkat V4.4
  - Lollipop от версии V5.0 до V5.1
  - Marshmellow V6.0
- Mobile Chrome для Google Android

При использовании HTML управления браузером в проекте WinCC веб-сервер поддерживает следующие панели Siemens HMI для стандартных страниц:

- Панели линейки Basic
  - Gen 2 KTP400 до KTP1200
- Панели линейки Comfort
  - от TP700 до TP2200
  - от KP400 до KP1500
  - KTP400
  - TP700 Comfort Outdoor
- Переносные панели
  - Gen 2 KTP700[F], KTP900[F]

Для получения информации о связанных с браузерами ограничениях, которые могут сказаться на отображении стандартных или пользовательских веб-страниц, смотри раздел Ограничения (Страница 1112).

Поведение веб-сервера в эксплуатации

Поведение веб-сервера в эксплуатации зависит от множества факторов. S7-1200 CPU и программатор должны совместно использовать время с другими задачами, которые потребляют ресурсы и время обработки. Если веб-сервер работает медленно, следует выполнить следующие настройки, чтобы улучшить производительность веб-сервера:

- Увеличить коммуникационную нагрузку (Страница 107) на PLC с 20 % до 50 %.
- Сконфигурировать минимальное время цикла (Страница 107). Установка минимального времени цикла обеспечить увеличение времени коммуникации между S7-1200 CPU и программатором.
- Вместо CP модуля (Страница 1033) для доступа к веб-серверу следует использовать Ethernet интерфейс S7-1200 CPU.

См. также

Буфер диагностики (Страница 1045)

12.1 Активация веб-сервера

Для активации веб-сервера в STEP 7 используется конфигурация устройства CPU, к которому должно быть выполнено подключение.

Для активации веб-сервера действовать следующим образом:

1. Выбрать CPU в конфигурации устройства.
2. В окне инспектора выбрать "Веб-сервер" в свойствах CPU.
3. Установить флажок для "Активировать веб-сервер на всех модулях этого устройства".
4. Выбрать усиленную защиту "Разрешен доступ только по протоколу HTTPS" для безопасного доступа к веб-серверу по умолчанию.
5. Если для "Автоматического обновления" выбрать опцию "Активировать автоматическое обновление", то стандартные веб-страницы будут обновляться каждые десять секунд. В поле "Интервал обновления" можно указать собственный период для обновления.

⚠ ПРЕДУПРЕЖДЕНИЕ

Несанкционированный доступ к CPU через веб-сервер

Несанкционированный доступ к CPU или установка недопустимых значений для переменных PLC могут нарушить ход процесса и привести к тяжелым телесным повреждениям и/или материальному ущербу.

Поскольку активация веб-сервера позволяет авторизованным пользователям выполнять изменения рабочего состояния, записывать данные PLC и обновлять прошивку, Siemens рекомендует придерживаться следующих мер безопасности:

- Активировать доступ к веб-серверу только по протоколу HTTPS.
- Защитите сложными паролями идентификаторы пользователей веб-сервера (Страница 1028). Сложные пароли - это, по крайней мере, десять символов, представляющих комбинацию букв, цифр и специальных символов, не являющиеся словами, которые могут быть найдены в словаре, и не являющиеся именами или идентификаторами, которые могут быть получены из персональных данных. Следует хранить пароль в секрете и часто его изменять.
- Не расширять минимальные права по умолчанию для пользователя веб-сервера "Все".
- Выполнить поиск ошибок и проверку диапазонов для переменных в программной логике, так как пользователи веб-страниц могут устанавливать для переменных PLC недопустимые значения.
- Использовать безопасную виртуальную частную сеть (VPN), чтобы подключаться к веб-серверу PLC S7-1200, находясь вне своей безопасной сети.

После загрузки конфигурации устройства, можно использовать стандартные веб-страницы, чтобы получить доступ к вводной и начальной странице CPU. Для доступа к другим страницам необходимо сконфигурировать одного или нескольких пользователей веб-сервера (Страница 1028).

Если создать и активировать пользовательские веб-страницы (Страница 1070), то можно получить доступ к ним из меню навигации стандартных или базовых веб-страниц.

Примечание

Замена устройства: Замена CPU V3.0 на CPU V4.x.x

При замене существующего CPU V3.0 на CPU V4.x (Страница 1655) и преобразовании своего проекта V3.0 в проект версии V4.x, следует помнить, что STEP 7 и CPU V4.x сохраняют следующие параметры настройки для веб-сервера:

- "Активировать веб-сервер на всех модулях этого устройства"
- "Разрешен доступ только по протоколу HTTPS"

Примечание

Если выполняется "Загрузка в RUN" (Страница 1410), то стандартные и пользовательские веб-страницы не обновляют значения данных и не разрешают изменять значения данных, пока загрузка не завершена. Веб-сервер отклоняет любые попытки записать значения данных во время выполнения загрузки.

12.2 Конфигурирование пользователей веб-сервера

Можно настроить пользователей с различными уровнями доступа для обращения к CPU через веб-сервер.

Для конфигурирования пользователей веб-сервера и соответствующих прав выполнить следующие действия:

1. Выбрать CPU в конфигурации устройства.
2. В окне инспектора выбрать "Веб-сервер" в свойствах CPU и активировать веб-сервер (Страница 1026).
3. В свойствах веб-сервера выбрать область "Управление доступом пользователей".
4. Ввести имена пользователей, уровни доступа и пароли для учетных записей пользователей, которые должны быть созданы.

После загрузки конфигурации в CPU, только авторизованные пользователи смогут получить доступ к функциям веб-сервера, для которых у них есть права.

Уровни доступа для веб-сервера

STEP 7 предлагает стандартного пользователя "Все" без пароля. По умолчанию этот пользователь не имеет никаких дополнительных прав и может просматривать только стандартные веб-страницы Старт (Страница 1042) и Введение (Страница 1041). Но можно предоставить дополнительные права пользователю "Все", а также другим пользователям:

- Диагностика посредством запросов
- Чтение переменных (тегов)
- Запись переменных (тегов)
- Чтение состояния переменных (тегов)
- Запись состояния переменных (тегов)
- Чтение пользовательских веб-страниц
- Запись в пользовательские веб-страницы
- Чтение файлов
- Запись/удаление файлов
- Изменение рабочего состояния
- Включение мигания светодиодов
- Выполнение обновления прошивки
- Создание резервной копии CPU
- Восстановление CPU из резервной копии
- Изменение системных параметров
- Изменение прикладных параметров

Если определенная пользователем веб-страница (Страница 1089) была определена в качестве страницы входа для веб-сервера, то у пользователя "Все" должно быть право "Чтение пользовательских веб-страниц".

⚠ ПРЕДУПРЕЖДЕНИЕ

Доступ к веб-серверу

Предоставление прав пользователю "Все" позволяет входит на веб-сервер без пароля. Несанкционированный доступ к CPU или установка недопустимых значений для переменных PLC могут нарушить ход процесса и привести к тяжелым телесным повреждениям и/или материальному ущербу.

Поскольку пользователь "Все" с достаточными правами может выполнять изменения рабочего состояния, записывать данные PLC и обновлять прошивку, Siemens рекомендует придерживаться следующих мер безопасности:

- Активировать доступ к веб-серверу только по протоколу HTTPS.
- Защитите сложными паролями идентификаторы пользователей веб-сервера. Сложные пароли - это, по крайней мере, десять символов, представляющих комбинацию букв, цифр и специальных символов, не являющиеся словами, которые могут быть найдены в словаре, и не являющиеся именами или идентификаторами, которые могут быть получены из персональных данных. Следует хранить пароль в секрете и часто его изменять.
- Не расширять минимальные права по умолчанию для пользователя веб-сервера "Все".
- Выполнить поиск ошибок и проверку диапазонов для переменных в программной логике, так как пользователи веб-страниц могут устанавливать для переменных PLC недопустимые значения.
- Использовать безопасную виртуальную частную сеть (VPN), чтобы подключаться к веб-серверу PLC S7-1200, находясь вне своей безопасной сети.

12.3 Доступ к веб-страницам от PC

Доступ к стандартным веб-страницам S7-1200 из PC или из мобильного устройства возможен с помощью IP-адреса S7-1200 CPU или IP-адреса активированного через веб-сервер CP (Страница 1033) в локальной стойке.

Чтобы получить доступ к стандартным веб-страницам S7-1200 из PC, выполнить следующие шаги:

1. Убедиться, чтобы S7-1200 и PC находятся в общей Ethernet-сети или соединены друг с другом напрямую с помощью стандартного Ethernet-кабеля.
2. Открыть веб-браузер и ввести URL "https://ww.xx.yy.zz". При этом "ww.xx.yy.zz" соответствует IP-адресу S7-1200 CPU или IP-адресу CP в локальной стойке.

Веб-браузер открывает стандартную вводную страницу (Страница 1041) или стандартную HTML страницу из пользовательских веб-страниц, если она была определена в качестве страницы входа (Страница 1089).

Примечание

Использовать безопасную виртуальную частную сеть (VPN), чтобы подключаться к веб-серверу PLC S7-1200, находясь вне своей безопасной сети. Кроме этого, следует учитывать все ограничения (Страница 1112), которые могут быть обусловлены веб-средой или операционной системой.

Доступ к стандартным веб-страницам через ввод URL страницы

Доступ к определенной стандартной веб-странице возможен через URL страницы. Для этого необходимо ввести URL в формате "https://ww.xx.yy.zz/<страница>.html", где <страница> соответствует IP-адресу S7-1200 CPU или IP-адресу CP в локальной стойке:

- https://ww.xx.yy.zz/start.html - Начальная страница (Страница 1042) с общей информацией о CPU
- https://ww.xx.yy.zz/identification.html - Идентификация (Страница 1043) с подробной информацией о CPU, включая серийный/заказной номера и номер версии, сейчас это страница диагностики
- https://ww.xx.yy.zz/module.html - Информация о модулях в локальной стойке и о возможностях для обновления прошивки (Страница 1046)
- https://ww.xx.yy.zz/communication.html - Коммуникационная информация (Страница 1050) о сетевых адресах, физических свойствах коммуникационных интерфейсов и коммуникационной статистике
- https://ww.xx.yy.zz/diagnostic.html - Буфер диагностики (Страница 1045)
- https://ww.xx.yy.zz/variable.html - Переменные CPU (теги) и ввод-вывод (Страница 1055), доступ по адресам или по имени переменной (тега) PLC/блока данных
- https://ww.xx.yy.zz/watch.html - Таблицы текущего контроля (Страница 1056)
- https://ww.xx.yy.zz/filebrowser.html- браузер для доступа к файлам архива данных или файлам рецептов (Страница 1068), хранящимся в CPU или на карте памяти
- https://ww.xx.yy.zz/index.html -Вводная страница (Страница 1041) для перехода к стандартным веб-страницам
- https://ww.xx.yy.zz/login.html - Страница для входа в систему (Страница 1038), если на данный момент нет пользователей в системе; в остальных случаях страница не содержит информации.

Например, если ввести "https://ww.xx.yy.zz/communication.html", браузер отображает страницу коммуникаций.

Примечание

Следует помнить, что каждая стандартная веб-страница, не указанная выше (напр., страница "Резервное онлайн-копирование" (Страница 1059)) не предлагает URL для прямого доступа.

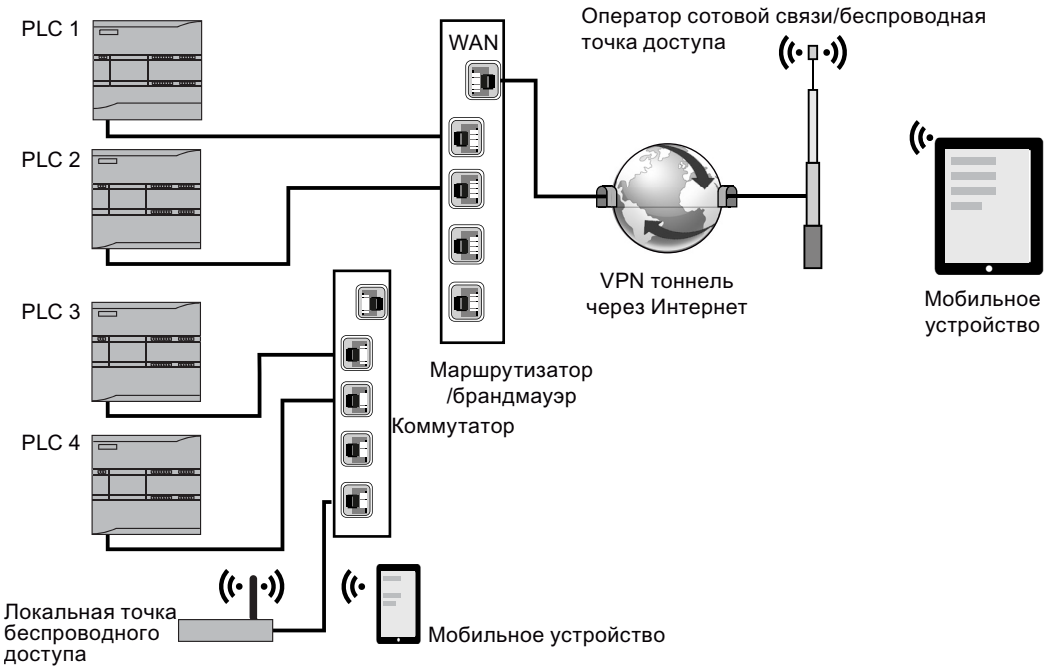
Защищенный доступ

Использовать безопасную виртуальную частную сеть (VPN), чтобы подключаться к веб-серверу PLC S7-1200, находясь вне своей безопасной сети. Следует установить по умолчанию и использовать https:// вместо http:// для защищенного доступа (Страница 1026) к стандартным веб-страницам. При установке соединения с S7-1200 посредством https://, веб-сайт шифрует сессию с помощью цифрового сертификата. Веб-сервер передает данные по защищенному соединению, и они недоступны для просмотра кем-либо. Как правило, отображается предупреждение системы безопасности, которое необходимо подтвердить с "Да" для перехода к стандартным веб-страницам. Чтобы отключить предупреждения системы безопасности при каждом защищенном доступе, можно импортировать сертификат ПО Siemens в свой веб-браузер (Страница 1033).

12.4 Доступ к веб-страницам через мобильное устройство

Чтобы получить доступ к S7-1200 из мобильного устройства, необходимо подключить PLC к сети, которая соединяется с Интернетом, или к локальной беспроводной точке доступа. Следует использовать безопасную виртуальную частную сеть (VPN) для подключения мобильного устройства к веб-серверу S7-1200 PLC. Можно использовать переадресацию портов в беспроводном маршрутизаторе, чтобы привязать IP-адрес PLC к адресу, по которому мобильное устройство может получить доступ к нему из Интернета. Для настройки переадресации портов, следовать инструкциям по конфигурированию ПО используемого маршрутизатора. Может быть подключено любое поддерживаемое маршрутизатором число PLC и коммутационной аппаратуры.

Без переадресации портов можно соединиться с PLC только локально в пределах зоны действия радиосигнала.



В этом примере мобильное устройство, которое находится в зоне действия локальной беспроводной точки доступа, может соединиться с PLC 3 и PLC 4 по их IP-адресам. Через интернет, вне зоны действия локальной беспроводной точки доступа, мобильное устройство может соединиться с PLC 1 и PLC 2, используя переадресацию для соответствующего PLC.

Для обращения к стандартным веб-страницам, необходимо иметь доступ к сотовой сети или точке беспроводного доступа. Чтобы получить доступ к PLC из Интернета, ввести в веб-браузере мобильного устройства адрес переадресации порта для доступа к PLC, например http: //ww.xx.yy.zz: pppp или https: //ww.xx. yy.zz: pppp, где ww.xx.yy.zz - адрес маршрутизатора, а pppp - назначение порта для конкретного PLC.

Для локального обращения через точку беспроводного доступа ввести IP-адрес S7-1200 CPU или IP-адрес активированного через веб-сервер CP (Страница 1033) в локальной стойке.

- http://ww.xx.yy.zz oder https://ww.xx.yy.zz для доступа к стандартным веб-страницам (Страница 1036)
- http://ww.xx.yy.zz/basic oder https://ww.xx.yy.zz/basic для доступа к базовым веб-страницам (Страница 1037)

Для повышенной безопасности сконфигурировать только защищенный доступ (HTTPS) (Страница 1026) к веб-серверу.

12.5 Использование CP модуля для доступа к веб-страницам

Независимо от того, выполняется ли доступ к веб-серверу с помощью PC или мобильного устройства, можно открывать стандартные веб-страницы через один из следующих CP модулей, если он был сконфигурирован в STEP 7 и установлен в локальную стойку с S7-1200 CPU:

- CP 1242-7 GPRS V2
- CP 1243-1
- CP 1243-7 LTE-EU
- CP 1243-7 LTE-US
- CP 1243-8 IRC

Для обращения к веб-страницам через эти CP модули используется стандартная начальная веб-страница (Страница 1042). Начальная страница отображает все сконфигурированные и установленные CP модули в локальной стойке, но доступ к веб-страницам возможен только через упомянутые выше модули.

Примечание

Доступ к стандартным веб-страницам, когда активированные через веб-сервер CP находятся в локальной стойке

При подключении к стандартным веб-страницам S7-1200 возможны задержки до одной или двух минут, когда активированные через веб-сервер CP находятся в локальной стойке. Если покажется, что страницы недоступны, или при появлении сообщений об ошибках, следует подождать одну или две минуты и обновить страницу.

12.6 Загрузка и установка сертификата безопасности

Можно загрузить стандартный сертификат безопасности Siemens в свои настройки Интернета.

Сертификат позволяет исключить проверку безопасности при каждом вызове https:// ww.xx.yy.zz, где «ww.xx.yy.zz» - это IP-адрес устройства, в веб-браузере. Если используется URL-адрес http://, а не https://, то загрузка и установка сертификата не требуется.

Начиная со STEP 7 версии V15 SP1 с поддержкой для S7-1200 CPU V4.3, можно создавать сертификаты в конфигурации устройства S7-1200 CPU. Эта функция доступна в общих настройка "Защита и безопасность > Администратор сертификатов" для устройства. В информационной системе STEP 7 можно найти инструкции об администраторе сертификации и о том, как создавать глобальные и локальные сертификаты для конкретного CPU.

Кроме этого, начиная со STEP 7 версии V15 SP1 с поддержкой для S7-1200 CPU V4.3, можно создавать сертификаты и для веб-сервера S7-1200 CPU.

Загрузка сертификата

Кликнуть на вводной странице (Страница 1041) по ссылке "Загрузить сертификат" для загрузки сертификата безопасности Siemens на свой PC. Порядок действий при загрузке и импорте зависит от используемого веб-браузера.

Импорт сертификата в Internet Explorer

1. Кликнуть на вводной странице по ссылке "Загрузить сертификат".
2. Нажать в появившемся диалоговом окне на "Открыть", чтобы открыть файл.
3. Нажать в диалоговом окне "Сертификат" на кнопку "Установить сертификат" для вызова мастера для импорта сертификата.
4. Нажать в диалоговом окне "Мастер импорта сертификатов" на "Дальше", чтобы настроить хранилище сертификатов.
5. Выбрать "Сохранять все сертификаты в следующее хранилище" и нажать кнопку "Просмотр".
6. В диалоговом окне "Выбрать хранилище сертификатов" выбрать "Third-Party Root Certification Authorities" и нажать OK.
7. Нажать на "Дальше" и после на "Готово", чтобы завершить работу "Мастера импорта сертификатов".

Импорт сертификата в Mozilla Firefox

1. Кликнуть на вводной странице по ссылке "Загрузить сертификат".
2. При появлении запроса, нажать "OK" в качестве подтверждения доверия к S7-1200 Controller Family.

В старых версиях Mozilla Firefox после выбора "Загрузить сертификат" необходимо сохранить файл и запустить мастера:

1. Кликнуть в диалоговом окне с открытым сертификатом по "Сохранить файл". Откроется диалоговое окно "Загрузки".
2. Двойной клик в диалоговом окне "Загрузки" по "MiniWebCA_Cer.crt" или имени созданного сертификаты. Если загрузка запускалась несколько раз, то отображается несколько копий. Двойной клик по любому из повторяющихся записей сертификата.
3. Нажать кнопку "OK", если необходимо подтвердить, что открывается исполняемый файл.
4. При необходимости выбрать "Открыть" в диалоговом окне "Открыть файл - предупреждение системы безопасности". Откроется диалоговое окно "Сертификат".
5. Нажать в диалоговом окне "Сертификат" на кнопку "Установить сертификат".
6. Придерживаться диалоговых окон "Мастера импорта сертификатов", чтобы импортировать сертификат. При этом операционная система автоматически выбирать хранилище сертификатов
7. При появлении диалогового окна "Предупреждение системы безопасности" нажать "Да" для подтверждения установки сертификата.

Другие браузеры

Придерживаться рекомендаций используемого веб-браузера для импорта и установки сертификата Siemens.

После установки сертификата безопасности Siemens "S7-1200 Controller Family" в параметрах Интернета для содержимого веб-браузера, больше не нужно проверять запрос безопасности при обращении к веб-серверу с помощью https: // ww.xx.yy. zz.

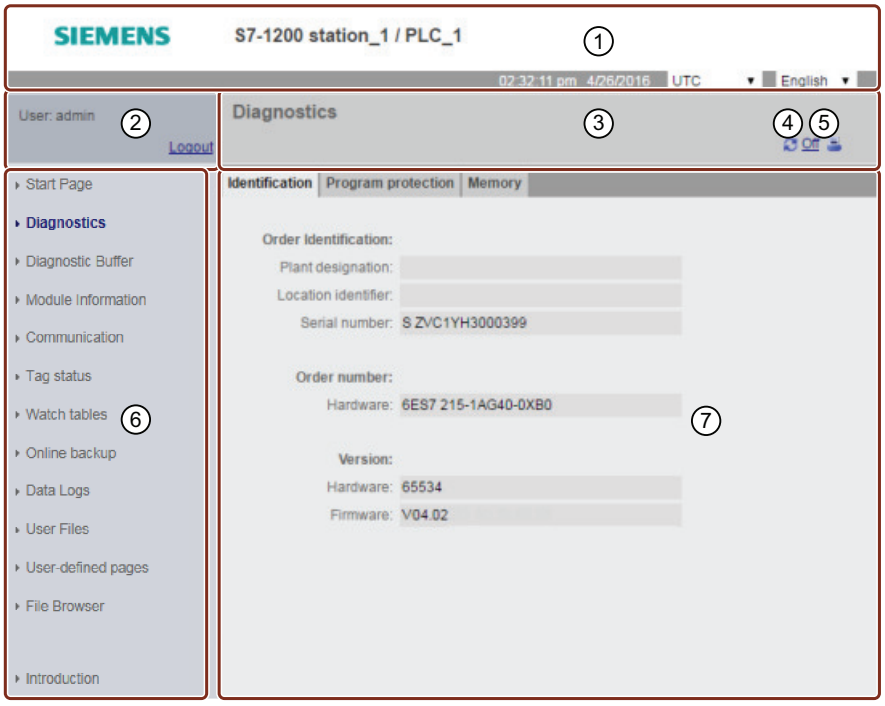
Примечание

Сертификат безопасности продолжает действовать и после перезапуска CPU. Но если IP-адрес устройства изменяется, то необходимо загрузить новый сертификат, если используется браузер, отличный от Internet Explorer или Mozilla Firefox.

12.7 Стандартные веб-страницы

12.7.1 Компоновка стандартных веб-страниц

Все стандартные веб-страницы S7-1200 имеют общую компоновку со ссылками для навигации и элементами управления для страницы. Независимо от того, открывается ли страница на PC или на мобильном устройстве, ее содержимое не изменяется. Но расположение и элементы навигации варьируются в зависимости от размера экрана и разрешении устройства. На стандартном PC или большом мобильном устройстве структура стандартной веб-страницы выглядит следующим образом:



① Заголовок веб-сервера с селектором для выбора локального времени PLC или времени UTC и языка отображения (Страница 167)

② Вход в систему или выход из системы

③ Строка заголовка стандартной веб-страницы с названием отображаемой страницы. В данном примере речь идет о странице с идентификацией CPU. На некоторых стандартных веб-страницах, напр., странице с информацией о модуле, здесь также отображают путь, если можно вызвать несколько страниц такого типа.

④ Символ для обновления: для страниц с автоматическим обновлением, разрешает или запрещает функцию автоматического обновления; для страниц без автоматического обновления, вызывает обновление страницы с текущими данными

⑤ Символ для печати: готовит и показывает версию для печати информации, доступной на отображаемой странице

⑥ Область навигации для перехода на другую страницу

⑦ Область содержимого для определенной стандартной веб-страницы, которая открыта в настоящий момент. В данном примере речь идет о странице диагностики.

Примечание

Стандартные веб-страницы CP модулей

Определенные CP модули (Страница 1033) предлагают стандартные веб-страницы, схожие по внешнему виду и функциям со стандартными веб-страницами S7-1200 CPU. Описание стандартных веб-страниц CP можно найти в документации CP модуля.

12.7.2 Базовые страницы

Веб-сервер предлагает базовые страницы для использования на мобильных устройствах. Для вызова базовых страниц используется IP-адрес устройства с "basic", прикрепленным к URL: http://ww.xx.yy.zz/basic или https://ww.xx.yy.zz/basic

Базовые страницы выглядят как стандартные веб-страницы, но с некоторыми отличиями. Страница не содержит области навигации, авторизации и заголовка, и содержит кнопки для перехода назад и вперед по веб-страницам. Кроме этого, на базовых страницах есть кнопка домашней страницы для перехода на страницу навигации. Но можно использовать и элементы навигации мобильного устройства. Так, например, выглядит базовая страница диагностики с вертикальной ориентацией:

Минимальное разрешение для отображения базовой страницы составляет 240 x 240 пикселей.



Необходимо учитывать, что иллюстрации к стандартным веб-страницам в этой главе соответствуют отображению на стандартном PC. У большинства стандартных веб-страниц есть эквивалентные базовые страницы.

12.7.3 Вход в систему и права пользователя

На PC каждая из стандартных веб-страниц предлагает окно входа в систему, расположенное над областью навигации. По соображениям экономии пространства, базовые веб-страницы предлагают отдельную страницу авторизации. S7-1200 поддерживает многопользовательскую авторизацию с различными уровнями доступа (правами):

- Диагностика посредством запросов
- Чтение переменных (тегов)
- Запись переменных (тегов)
- Чтение состояния переменных (тегов)
- Запись состояния переменных (тегов)
- Чтение пользовательских веб-страниц
- Запись в пользовательские веб-страницы
- Чтение файлов
- Запись/удаление файлов
- Изменение рабочего состояния
- Включение мигания светодиодов
- Выполнение обновления прошивки
- Создание резервной копии PLC
- Восстановление PLC с помощью резервной копии прошивки

В конфигурации устройства CPU в STEP 7 в свойствах управления доступом пользователей для веб-сервера устанавливаются роли пользователей, назначаются уровни доступа и присваиваются пароли (Страница 1028).

Вход в систему

STEP 7 предлагает стандартного пользователя "Все" без пароля. По умолчанию этот пользователь не имеет никаких дополнительных прав и может просматривать только стандартные начальную (Страница 1042) и вводную (Страница 1041) веб-страницы. Но пользователю "Все" могут быть предоставлены дополнительные права, как и всем другим конфигурируемым пользователям:

⚠ ПРЕДУПРЕЖДЕНИЕ

Доступ к веб-серверу

Предоставление прав пользователю "Все" позволяет входит на веб-сервер без пароля. Несанкционированный доступ к CPU или установка неправильных значений для переменных PLC могут нарушить ход процесса и привести к тяжелым телесным повреждениям и/или материальному ущербу.

Поскольку пользователь "Все" с достаточными правами может выполнять изменения рабочего состояния, записывать данные PLC и обновлять прошивку, Siemens рекомендует придерживаться следующих мер безопасности:

- Активировать доступ к веб-серверу только по протоколу HTTPS.
- Защитить сложным паролем идентификаторы пользователей веб-сервера (Страница 1028). Сложные пароли - это, по крайней мере, десять символов, представляющих комбинацию букв, цифр и специальных символов, не являющиеся словами, которые могут быть найдены в словаре, и не являющиеся именами или идентификаторами, которые могут быть получены из персональных данных. Следует хранить пароль в секрете и часто его изменять.
- Не расширять предустановленные минимальные права для пользователя "Все".
- Выполнить поиск ошибок и проверку диапазонов для переменных в программной логике, так как пользователи веб-страниц могут устанавливать для переменных PLC недопустимые значения.
- Использовать безопасную виртуальную частную сеть (VPN), чтобы подключаться к веб-серверу PLC S7-1200, находясь вне своей безопасной сети.

Для выполнения определенных действий, напр., изменения рабочего режима контроллера, записи значений в память и обновления прошивки CPU, потребуются специальные права. Если для Степени защиты CPU (Страница 203) была выбрана опция

"Без доступа (полная защита)", то у пользователя "Все", независимо от установленных прав пользователя веб-сервера, нет доступа к веб-серверу.



Поле для входа в систему при отображении на PC или мобильном устройстве с большим дисплеем находится сверху слева на каждой стандартной веб-странице.



На компактных мобильных устройствах, отображающих базовые страницы, есть отдельная страница регистрации. Она может быть вызвана с начальной страницы.

Для входа в систему выполнить следующие действия:

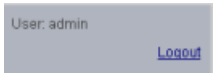
1. Ввести имя пользователя в поле "Имя пользователя".
2. Ввести пароль в поле "Пароль".

Если активность отсутствует в течение тридцати минут, выполняется автоматический выход и системы. Если отображаемая страница постоянно обновляется, то отсчета времени для выхода из системы не выполняется и сеанс продолжается.

Примечание

Если при входе в систему возникают ошибки, то следует загрузить сертификат безопасности Siemens (Страница 1033) с вводной страницы (Страница 1041). После можно войти в систему без ошибок.

Выход из системы





Для выхода из системы кликнуть на PC или на мобильном устройстве на ссылку "Выход из системы" на любой странице.

Перейти через базовые страницы с начальной страницы на страницу для входа и выхода и нажать не кнопку "Выход из системы".

После выхода из системы можно вызывать и просматривать стандартные веб-страницы согласно правам пользователя "Все". В описаниях отдельных стандартных веб-страниц определяются необходимые права для страницы.

Примечание

Выход из системы перед закрытием веб-сервера

Если был выполнен вход в систему на веб-сервере, то необходимо обязательно выйти из системы перед закрытием веб-браузера. Веб-сервер поддерживает максимум семь параллельных авторизаций.

12.7.4 Вводная страница

Вводная страница - это страница приветствия для перехода к стандартным веб-страницам S7-1200/



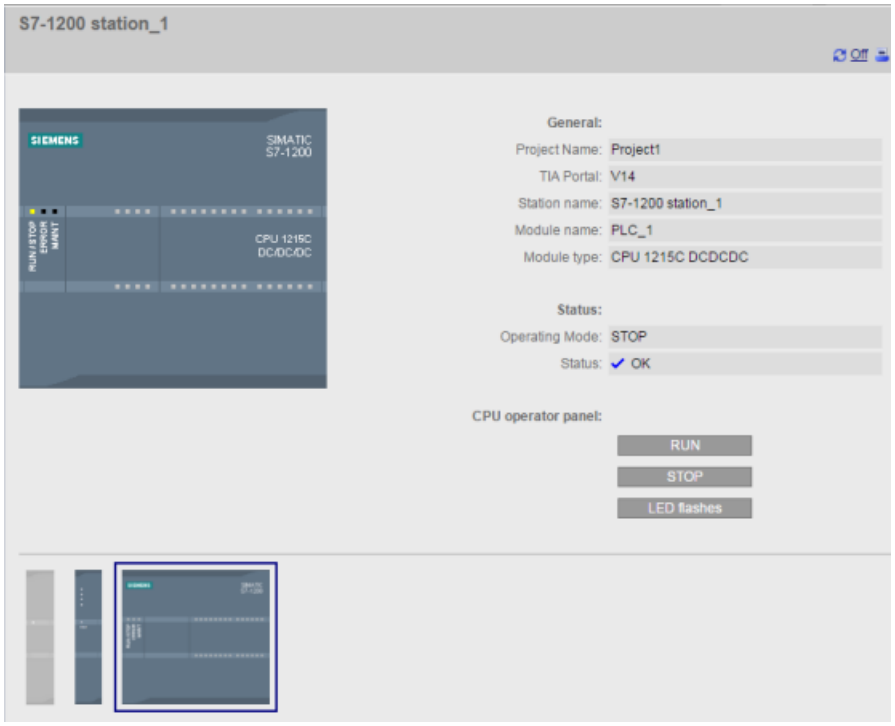
Нажать "Ввод" на этой станице для перехода на стандартные веб-страницы S7-1200. Сверху на странице находятся ссылки для перехода на полезный веб-ресурсы Siemens, а также ссылка для загрузки сертификата безопасности Siemens (Страница 1033). Также можно указать, что вводная страница должна быть пропущена при последующих обращениях к веб-серверу.

12.7.5 Начальная страница

Начальная страница содержит представление CPU или CP, к которому выполнено подключение, и общую информацию об устройстве, а также версию TIA Portal, с помощью которой проект был загружен в CPU. Для CPU с помощью кнопок можно изменять рабочее состояние и управлять светодиодными индикаторами, если вход в систему (Страница 1038) был выполнен с правом (Страница 1028) на изменение рабочего режима.

В нижней части экрана видно, сконфигурированы ли и установлены ли активированные через веб-сервер CP модули (Страница 1033) в локальной стойке с S7-1200 CPU. Для доступа к стандартным веб-страницам навести курсор на один из активированных через веб-сервер CP модулей и кликнуть по нему. Информацию о веб-страницах CP модулей можно найти в документации к конкретному CP модулю. Название CP модуля появляется после наведения на него курсора.

Веб-сервер показывает и другие CM и CP модули в локальной стойке, но кликнуть по ним нельзя, так как они не содержат веб-страниц. Эти CM и CP модули представлены в светло-сером цвете (недоступны для выбора), чтобы указать на то, что они только отображаются и не активируются по клику мыши.



Обратить внимание на то, что отказобезопасные S7-1200 CPU отображают на этой странице с дополнительными данными, относящимися к функциональной безопасности.

12.7.6 Диагностика

Страница диагностики содержит характеристики для идентификации CPU, настройки конфигурации для защиты ноу-хау и использования ресурсов загружаемой, рабочей и сохраняющей памяти:

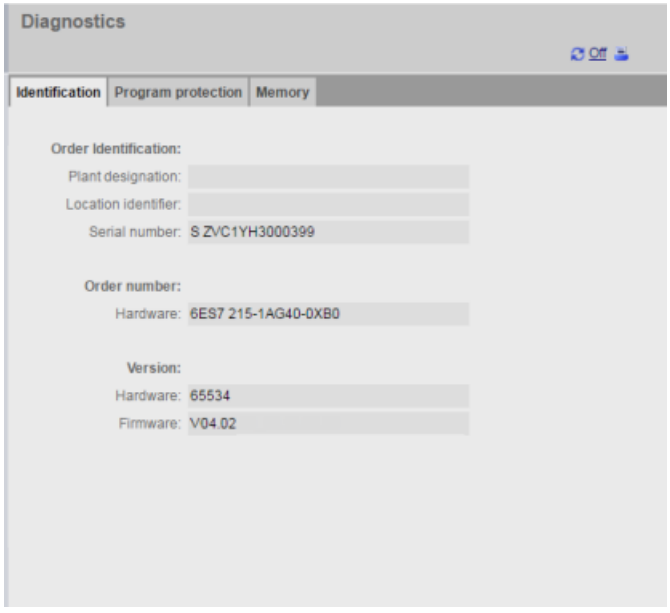
На странице имеется три вкладки:

- Идентификация: характеристики для идентификации модуля и установки и информации о местоположении из STEP 7
- Защита программы: состояние защиты ноу-хау и привязки CPU, которые могут быть полезны при планировании снабжения запасными частями, а также настройки конфигурации STEP 7 для разрешения или запрета копирования внутренней загружаемой памяти во внешнюю загружаемую память (карта памяти SIMATIC).
- Память: Степень использования загружаемой, рабочей и сохраняющей памяти

Для F-CPU предлагается дополнительная вкладка с параметрами отказоустойчивости.

Для отображение страницы идентификации потребуется право (Страница 1028) "Диагностика посредством запросов".

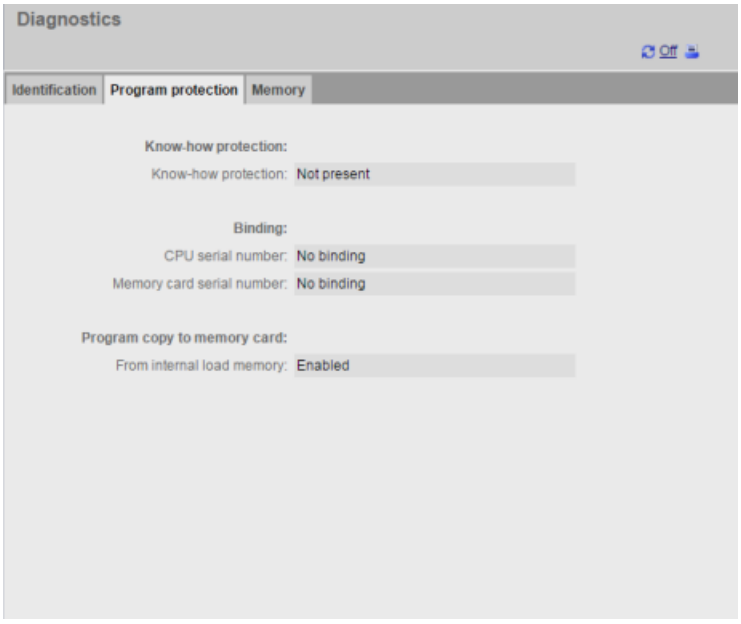
Вкладка "Идентификация"



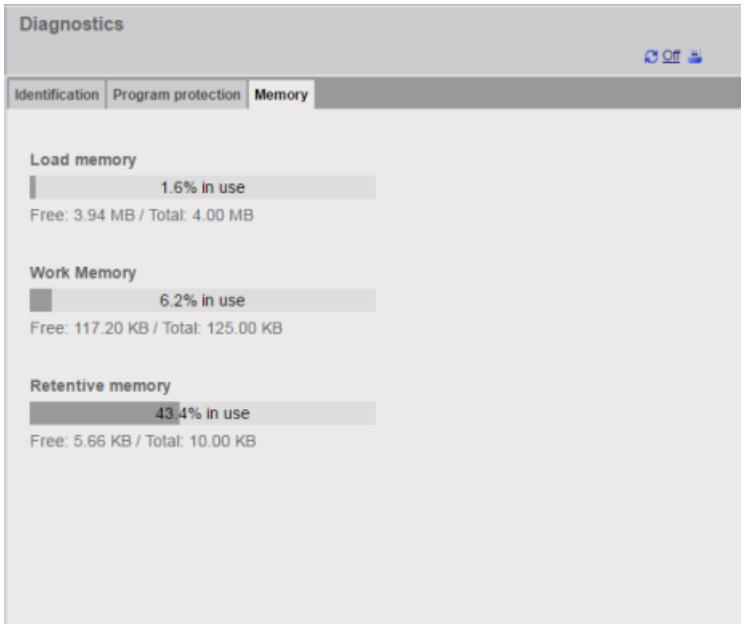
Вкладка "Защита программы"

Вкладка "Защита программы" содержит следующую информацию:

- Защита ноу-хау (Страница 206): Показывает, была ли сконфигурирована защиты ноу-хау для блоков программы в STEP 7.
- Привязка (Страница 207): Показывает, была ли программа привязана к CPU или к карте памяти SIMATIC.
- Копирование программы на карту памяти (Страница 206): Показывает, была ли активирована возможность для копирования программы из внутренней загружаемой памяти во внешнюю загружаемую память (карта памяти SIMATIC).



Вкладка "Память"



Вкладка "Отказоустойчивость"

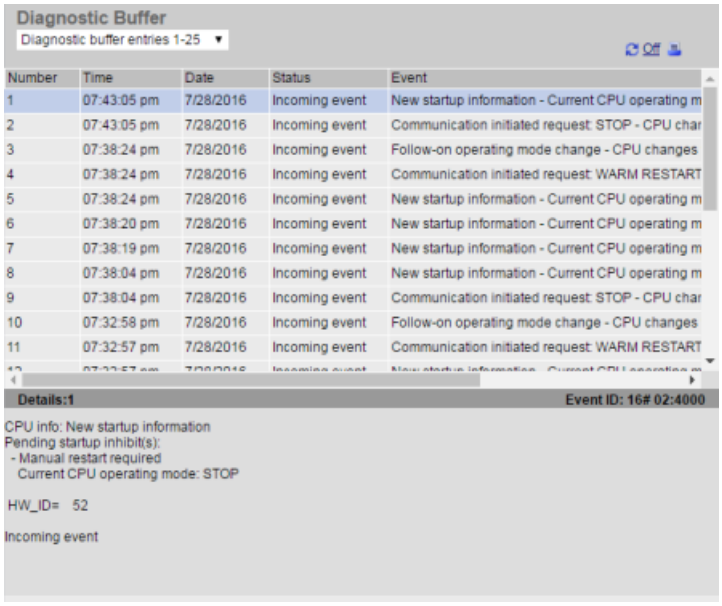
Дополнительную информацию о странице диагностики вкладки "Отказоустойчивость" можно найти вS7-1200 руководство по функциональной безопасности (https://support.industry.siemens.com/cs/ww/en/view/104547552).

12.7.7 Буфер диагностики

Страница буфера диагностики отображает диагностические события. Последнее событие отображается в самом верху под номером 1. Самое старое событие имеет номер 50. В левом селекторе можно выбрать, какой диапазон записей в буфере диагностики должен отображаться, записи от 1 до 25 или записи от 26 до 50. В правом селекторе можно выбрать, будет ли для отображения времени использоваться UTC или локальное время PLC. В верхней части страницы эти диагностические записи отображаются со временем и датой возникновения.

Можно выбрать любую отдельную диагностическую запись, чтобы показать подробную информацию об этой записи в нижней части страницы. Следует помнить, что язык отображения для записей буфера диагностики зависит от установленной конфигурации устройства для многоязыковой поддержки (Страница 172).

На страничке буфера диагностики также есть кнопка для сохранения буфера диагностики в файл CSV. По умолчанию веб-сервер сохраняет файл в формате с разделением элементов запятой как ASLog.csv в папку загрузки. Файл содержит весь буфер диагностики на момент сохранения. Можно сохранять файл столько раз, сколько необходимо, и хранить несколько файлов. Файл буфера диагностики можно открыть с помощью Microsoft Excel или любого текстового редактора.



Для отображения страницы буфера диагностики необходимо право (Страница 1028) на диагностику посредством запросов.

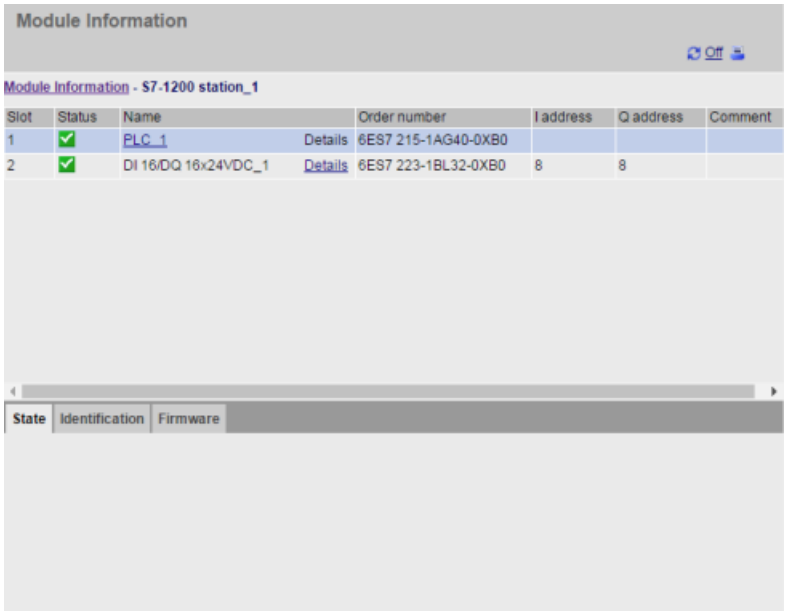
12.7.8 Информация о модулях

Страница "Информация о модулях" предоставляет информацию обо всех модулях в локальной стойке. В верхний части страницы отображается обзор модулей на основе конфигурации устройства в STEP 7, а в нижней части отображается состояние, идентификационные данные и информация о прошивке подключенного модуля. Страница "Информация о модулях" также позволяет выполнить обновление прошивки.

Для отображения страницы информации о модулях необходимо право (Страница 1028) на диагностику посредством запросов.

Информация о модулях: Вкладка "Состояние"

Вкладка "Состояние" в нижней части страницы информации о модулях показывает описание текущего состояния модуля, который выбран в верхней части. Если секция пуста, то модуль не имеет ожидающего диагностического состояния.



Символы состояния для модулей

Для каждого модуля в верхней части столбца состояния отображается символ, обозначающий состояние модуля:

Символ	Значение
	Неполадки отсутствуют
	Отключено
	Требуется техобслуживание
	Запрос на техобслуживание
	Ошибка
	CPU не может связаться с модулем или устройством (для устройств, отличных от CPU).
	CPU установил соединение с устройством, но состояние модуля неизвестно (для устройств, отличных от CPU).
	Данные по входам и выходам недоступны, так как субмодуль заблокировал свои каналы ввода-вывода (для устройств, отличных от CPU).

Отображение дополнительной информации

Можно выбрать ссылку в верхней части, чтобы показать дополнительную информацию об определенном модуле. У модулей с субмодулями есть ссылки для каждого субмодуля. Тип выводимой информации меняется в зависимости от выбранного модуля. Например, диалоговое окно информации о модуле первоначально отображает имя станции SIMATIC S7-1200, индикатор состояния и комментарий. Если развернуть ссылку до уровня CPU, то выводится информации об именах цифровых и аналоговых входов и выходов для конкретной модели CPU, информация об адресах ввода-вывода, индикаторы состояния, номера слотов и комментарии.

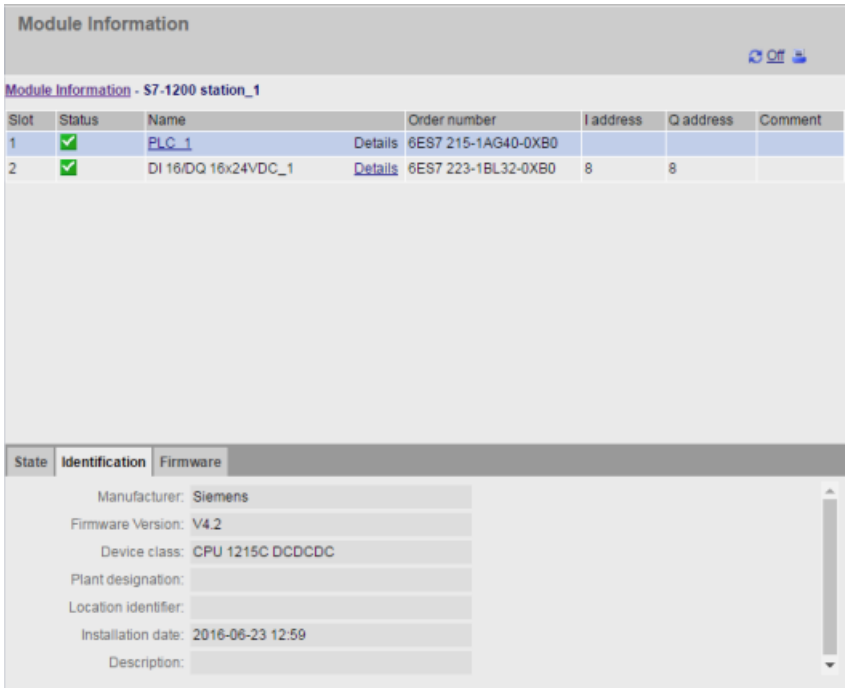
Slot	Status	Name		Order number	I address	Q address	Comment
1.16	✔	HSC_1	Details	6ES7 214-1AG40-0XB0	1000	---	
1.17	✔	HSC_2	Details	6ES7 214-1AG40-0XB0	1004	---	
1.18	✔	HSC_3	Details	6ES7 214-1AG40-0XB0	1008	---	
1.19	✔	HSC_4	Details	6ES7 214-1AG40-0XB0	1012	---	
1.20	✔	HSC_5	Details	6ES7 214-1AG40-0XB0	1016	---	
1.21	✔	HSC_6	Details	6ES7 214-1AG40-0XB0	1020	---	
1.2	✔	AI2_1	Details	6ES7 214-1AG40-0XB0	64	---	
1.1	✔	DI14/DQ10_1	Details	6ES7 214-1AG40-0XB0	0	0	
1.32	✔	Pulse_1	Details	6ES7 214-1AG40-0XB0	---	1000	

При развертывании экрана, страница информации о модулях показывает соответствующий путь. Можно кликнуть по любой ссылке на этом пути, чтобы возвратиться на более высокий уровень.



Информация о модулях: Вкладка "Идентификация"

На вкладке "Идентификация" отображается вся информация об идентификации и обслуживании (Identification and Maintenance, I&M) для выбранного модуля.



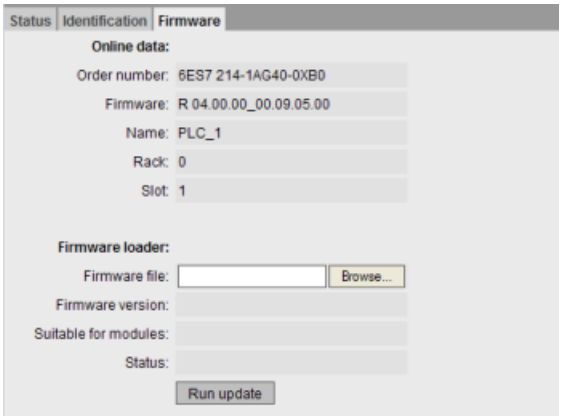
Кликнув в верхней части по F-модулю ввода-вывода, в нижней части откроется вкладка "Безопасность". На этой вкладке специальные данные по выбранному модулю отображаются так, как описано в S7-1200 руководство по функциональной безопасности (https://support.industry.siemens.com/cs/ww/en/view/104547552).

Информация о модулях: Вкладка "Прошивка"

На вкладке "Прошивка" страницы информации о модуле отображает информации о прошивке выбранного модуля. При наличии прав (Страница 1028) на обновление прошивки, можно выполнить обновление прошивки CPU или других модулей в локальной стойке, поддерживающих обновление прошивки, а также модулей PROFINET I/O. Для распределенных модулей можно просмотреть информацию о прошивке, но нельзя выполнить ее обновление.

Примечание

Обновление прошивки CPU возможно только для S7-1200 CPU от версии 3.0.



Выполнение обновления прошивки

Для выполнения обновления прошивки CPU должен находиться в рабочем состоянии STOP. При нахождении CPU в состоянии STOP, нажать кнопку "Просмотр" для выбора файла прошивки. Обновления прошивки находятся на сайте Онлайн-поддержки для промышленности Siemens (http://support.industry.siemens.com).

Во время обновления, страница отображает сообщение о выполнении обновления. После того, как обновление завершено, страница отображает заказной номер и номер версии обновленной прошивки. После обновления прошивки CPU или сигнальной платы, веб-сервер перезапускает CPU.

Обновление прошивки может быть выполнено и одним из следующих способов:

- С помощью Интерактивных и диагностических инструментов STEP 7 (Страница 1394)
- С помощью карты памяти SIMATIC (Страница 147)
- С помощью SIMATIC Automation Tool (https://support.industry.siemens.com/cs/ww/en/view/98161300)

Примечание

Возможные проблемы при выполнении обновления прошивки через веб-сервер

В случае нарушения связи во время обновления прошивки через веб-сервер, в веб-браузере может появиться сообщение с вопросом, покинуть или остаться на текущей странице. Во избежание потенциальных проблем, выбрать опцию, чтобы остаться на текущей странице.

Если закрыть веб-браузер во время выполнения обновления прошивки через веб-сервер, то не получится изменить режим работы CPU на RUN. Если это происходит, то необходимо выключить и снова включить CPU, чтобы перевести CPU в режим RUN.

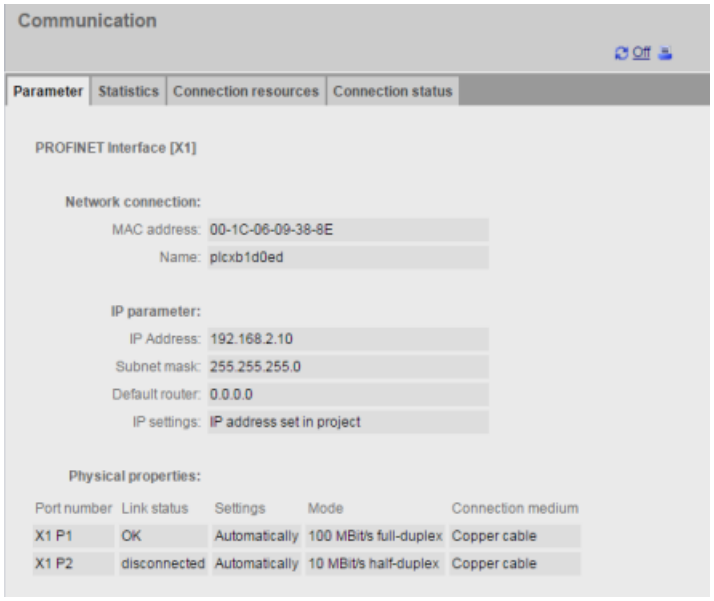
12.7.9 Коммуникация

Страница коммуникации показывает параметры подключенного CPU, статистику по коммуникации, ресурсы и информацию о соединениях.

Для отображение страницы коммуникации потребуется право "Диагностика посредством запросов".

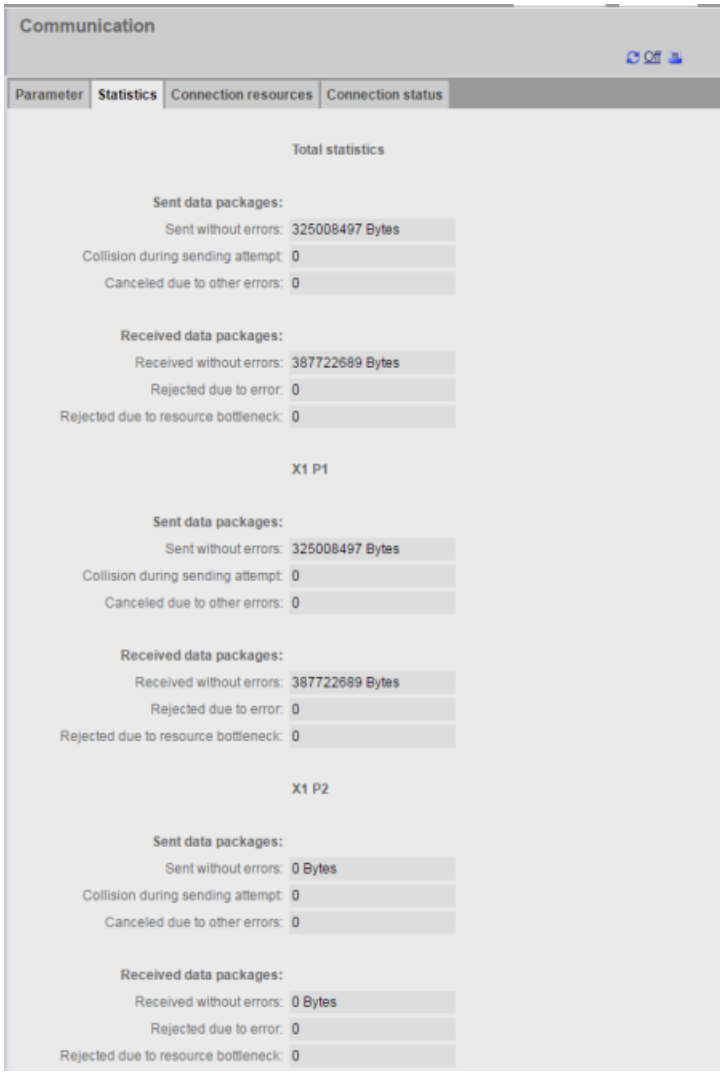
Вкладка "Параметры"

Вкладка "Параметры" показывает MAC-адрес CPU, IP-адрес и настройки IP для CPU, а также физические характеристики:



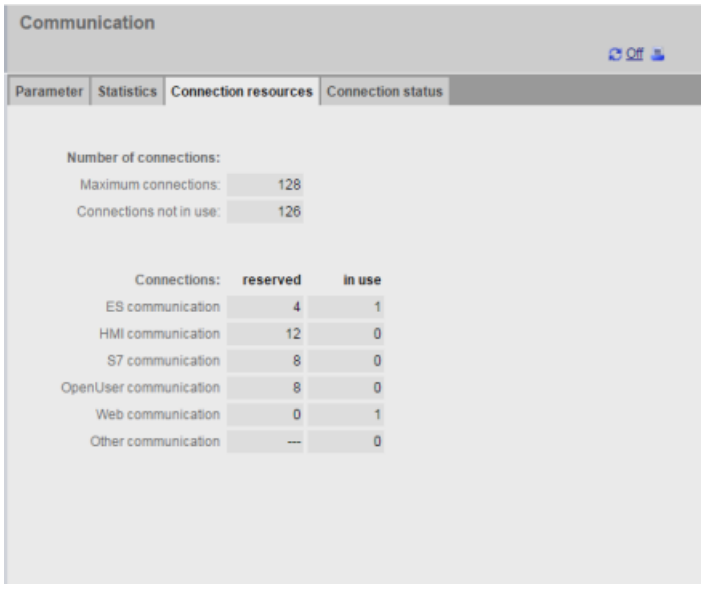
Вкладка "Статистика"

Вкладка "Statistik" показывает статистическую информацию по передаче и приему данных:



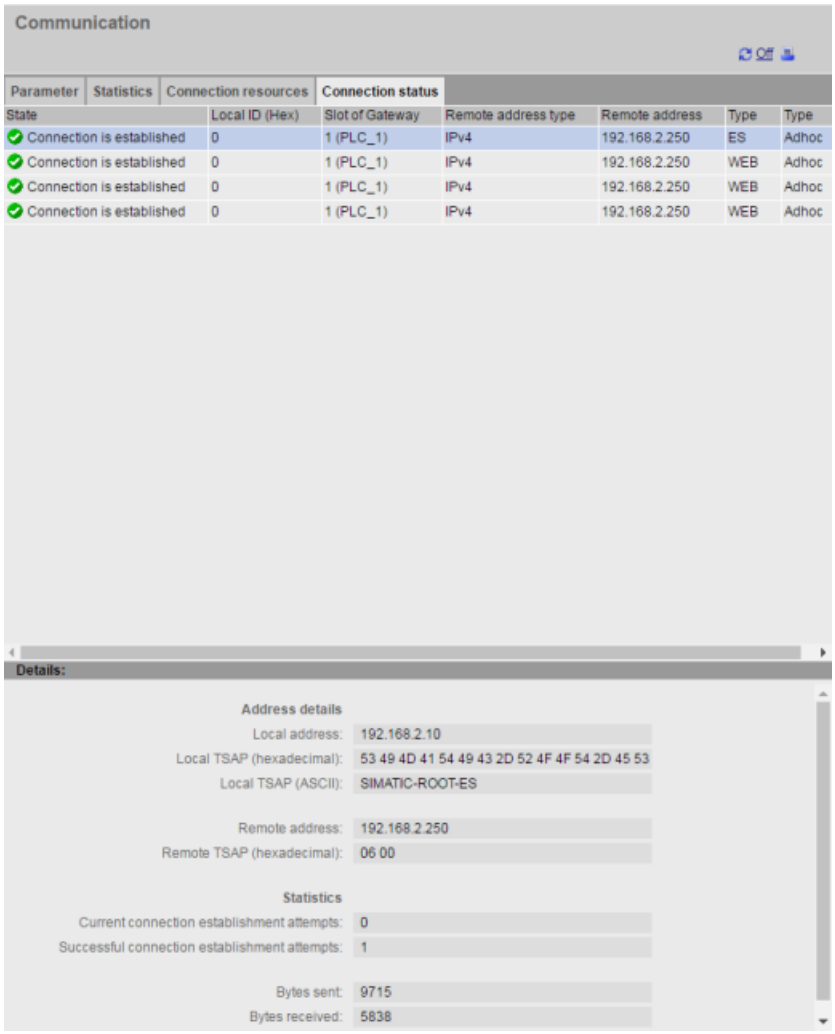
Вкладка "Ресурсы соединения"

Вкладка "Ресурсы соединения" отображает информацию об общем числе ресурсов соединения и как они распределены по различным типам коммуникации:



Вкладка "Состояние соединения"

Вкладка "Состояние соединения" показывает соединения CPU и подробности для выбранного соединения.

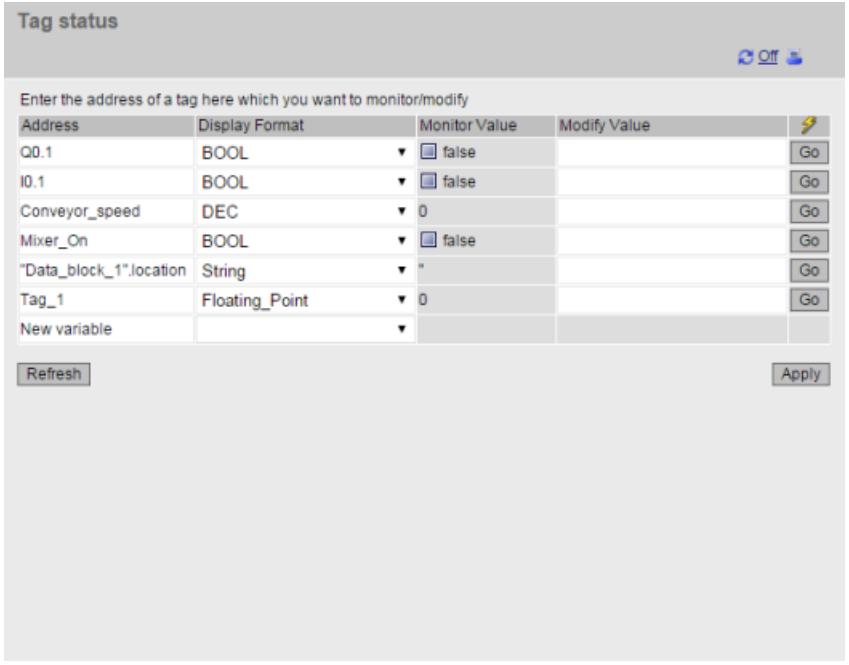


12.7.10 Состояние переменных (тегов)

На странице Состояния переменных (тегов) можно отобразить все входы и выхода или данные по памяти в CPU. Вы можете ввести прямой адрес (такой как I0.0), имя переменной (тега) PLC или переменную (тег) из определенного блока данных. Для переменных (тегов) блока данных имя блока заключается в двойные кавычки. Для каждого контролируемого значения можно выбрать формат отображения для данных. Можно продолжать вводить и определять значения до тех пор, пока это позволяют ограничения для страницы. Контролируемые значения отображаются автоматически. С помощью кнопки "Обновить" можно в любое время обновить все контролируемые значения. Если в STEP 7 было активировано автоматическое обновление (Страница 1026), можно кликнуть вверху справа на странице по "Выкл", чтобы отключит эту установку. Если автоматическое обновление отключено, то можно его снова включить с помощью "Вкл".

Для отображения страницы Состояние переменных (тегов) потребуется право "Чтение состояния переменных (тегов)".

После входа в систему в качестве пользователя с правом (Страница 1038) "Записи состояния переменных (тегов)", также можно изменять значения данных. Ввести в соответствующем поле "Измененное значение" значения, которые должны быть установлены. Нажать на кнопку "Выполнить" рядом со значением для записи значения в CPU. Можно ввести и несколько значений и нажать "Применить", чтобы записать все значения в CPU. Кнопки и метки столбцов для изменения отображаются только в том случае, если у пользователя есть право "Записи состояния переменных (тегов)".



Если уйти со страницы Состояние переменных (тегов) и снова вернуться на нее, то введенные на странице Состояние переменных (тегов) данные не будут сохранены. Можно сохранить страницу как закладку и после вернуться на нее, чтобы просматривать те же записи. Если страница не сохраняется в закладки, то потребуется повторно ввести переменные.

Для значений, которые часто контролируются или изменяются, следует рассмотрите возможность использования Таблицы текущего контроля (Страница 1056).

Примечание

Обратить внимание на следующие моменты при использовании стандартной страницы Состояние переменных (тегов):

- Заключать все измененные значения для строк в одиночные кавычки.
- Страница Состояние переменных (тегов) может контролировать или изменять переменные (теги), которые содержат любой из следующих символов: &, <, (, +, , (запятая), ., [,], $ или %, при условии заключения имени переменой (тега) в двойные кавычки, напр., "Clock_2.5Hz".
- Для контроля или изменения только одного поля DTL переменной (тега), следует включить поле в адрес, напр., "Data_block_1".DTL_tag.Year. Ввести целое число для измененного значения согласно типу данных определенного DTL поля. Пример: Поле года имеет тип данных UInt.
- Максимальное количество записей переменных (тегов) на страницу равно 50.
- Если имя переменной (тега) содержит специальные символы и поэтому имя отклоняется как элемент на странице Состояние переменных (тегов), можно заключить имя переменной (тега) в двойные кавычки. После этого в большинстве случаев страница определяет имя переменной (тега).

См. также

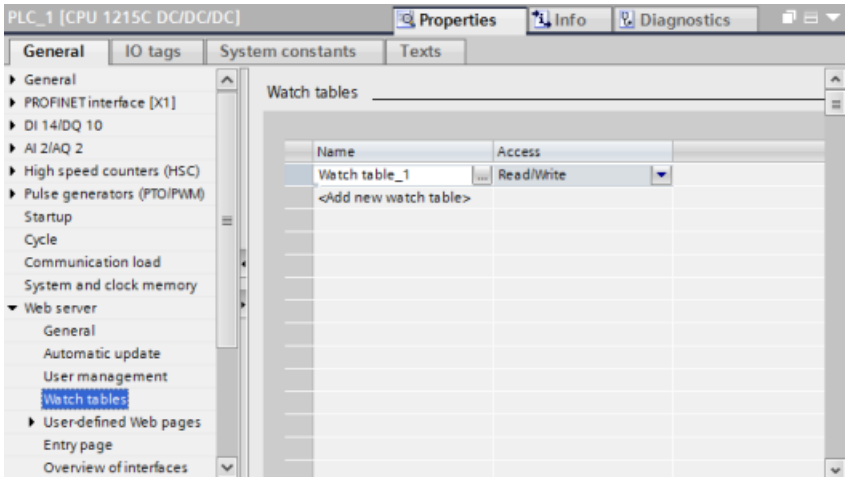
Правила ввода имен переменных (тегов) и значений (Страница 1114)

12.7.11 Таблицы текущего контроля

Через веб-сервер возможен доступ к таблицам текущего контроля, сконфигурированным в STEP 7 и загруженным в CPU. Таблицы текущего контроля с 50 или меньшим числом элементов обеспечивают наилучшую производительность в веб-сервере.

Конфигурация STEP 7 для выбора таблиц текущего контроля для веб-сервера

В конфигурации устройства в STEP 7 можно добавлять таблицы текущего контроля, которые должны отображаться веб-сервером. Для каждой таблицы текущего контроля, выбираемой в списке имеющихся таблиц текущего контроля, назначаются либо права на чтение, либо права на чтение и запись. После загрузки в CPU, таблицы текущего контроля, которым были присвоены права на чтение, будут доступны только для просмотра, а в таблицах текущего контроля с правом на чтение и запись можно будет отображать и изменять переменные (теги).



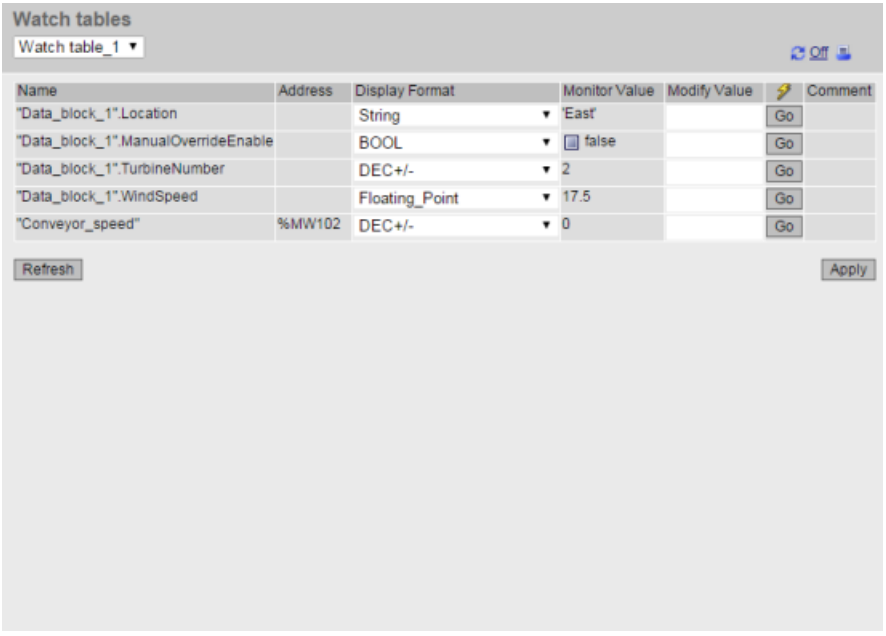
После завершения конфигурирования таблицы текущего контроля в области "Веб-сервер" конфигурации устройства, следует загрузить конфигурацию оборудования в CPU.

Просмотр таблиц текущего контроля через веб-сервер

При наличии права (Страница 1028) на "чтение переменных (тегов)" в навигационном меню можно выбирать "Таблицы текущего контроля" для обращения к сконфигурированным и загруженным в CPU таблицам текущего контроля. Если было загружено несколько таблиц текущего контроля загружено, выбрать в выпадающем списке таблицу, которая должна быть отображена. Веб-сервер показывает таблицу текущего контроля, созданную в STEP 7, и текущие значения согласно формату отображения. Формат отображения может изменяться произвольно, но при возвращении на страницу таблиц текущего контроля веб-сервер по умолчанию снова использует форматы отображения в таблице текущего контроля STEP 7.

Изменение переменных (тегов) в таблицах текущего контроля через веб-сервер

После загрузки таблицы текущего контроля с уровнем доступа "Чтение/запись" и входа на веб-сервер с правом (Страница 1028) "Записи переменных (тегов)", можно изменять значения переменных (тегов) точно так же, как в таблице текущего контроля в STEP 7. Можно изменить одно единственное значение переменной (тега) и нажать на "Выполнить", чтобы изменить только это значение, или можно ввести несколько значений и нажать на "Применить", чтобы за один раз изменить все значения.



Примечание

Преимущества таблиц текущего контроля для изменения переменных (тегов)

Для того, чтобы пользователь смог изменять переменные (теги) и переменные (теги) блоков данных в CPU через таблицу текущего контроля, необходимо сконфигурировать таблицу текущего контроля в свойствах веб-сервера в конфигурации устройства STEP 7, а также присвоить таблице доступ по чтению и записи. Тем самым набор переменных (тегов), которые сможет изменять пользователь с правом "Запись переменных (тегов)", будет ограничен переменными (тегами) в сконфигурированных таблицах текущего контроля веб-сервера.

С другой стороны, на странице Состояние переменных (тегов) (Страница 1055) любой пользователем с правом "Запись состояния переменных (тегов)" сможет выполнять запись в любую переменную (тег) или переменную (тег) блока в CPU.

Путем тщательного конфигурирования прав в управлении пользователями (Страница 1028) веб-сервера можно защитить доступ к своим данным PLC.

См. также

Правила ввода имен переменных (тегов) и значений (Страница 1114)

12.7.12 Резервное онлайн-копирование

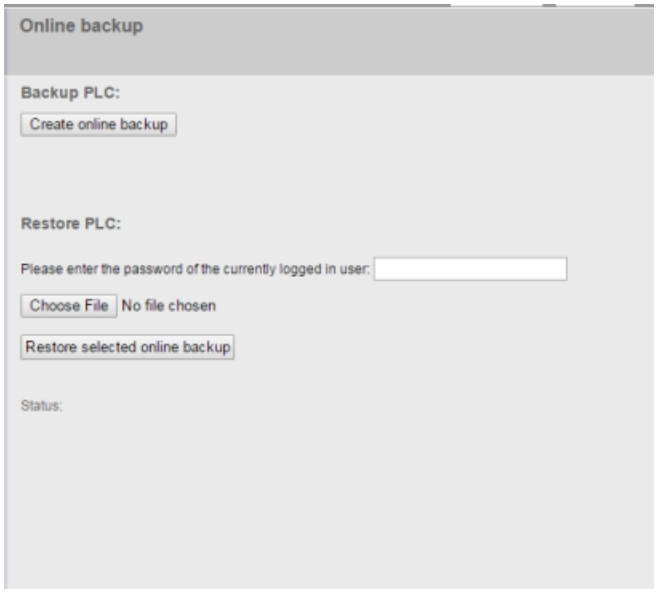
На стандартной веб-странице Резервное онлайн-копирование можно создать резервную копию проекта STEP 7 для Онлайн-PLC или восстановить созданную прежде резервную копию PLC. Перед созданием или восстановлением резервной копии следует перевести PLC в рабочее состояние STOP и завершить любую коммуникацию с PLC, как то доступ через HMI и доступ к веб-серверу. Если CPU не находится в состоянии STOP, то функции резервного копирования и восстановления выводят сообщение с требованием на перевод CPU в состояние STOP для продолжения работы.

Если страница "Резервное онлайн-копирование" была вызвана через один из CP модулей с веб-поддержкой, то возможно резервное копирование, но не восстановление.

Примечание

Функции резервного копирования и восстановления доступны и в STEP 7 (Страница 1426). Из полного описания этой темы можно узнать, какие данные могут быть включены в резервную копию и восстановлены. Инструментарий SIMATIC Automation Tool (SAT) также предлагает функции резервного копирования и восстановления.

При резервном копировании файлов через веб-сервер, PC или устройство сохраняет резервные файлы в стандартную парку для загрузки. При резервном копировании файлов через STEP 7 они сохраняются в проект STEP 7. Нельзя восстановить резервные файлы STEP 7 через веб-сервер, как нельзя восстановить и резервные файлы веб-сервера через STEP 7. Но можно сохранить резервные файлы STEP 7 напрямую в папку загрузки PC или устройства. Если действовать таким образом, то файлы смогут быть восстановлены через веб-сервер.



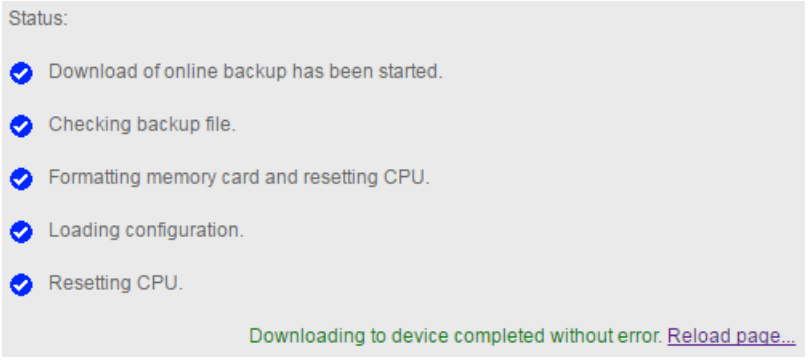
Резервная копия PLC

Нажать в области Резервная копия PLC страницы на кнопку "Создать резервную онлайн-копию", чтобы создать резервную копию для проекта, который в настоящее время сохранен в PLC. Для этой функции потребуется право (Страница 1028) "Резервное копирование CPU". Если CPU находится в рабочем состоянии RUN и должен быть переведен в состояние STOP, дополнительно потребуется и право "Изменение рабочего состояния". PC или устройство сохраняет резервный файл в стандартную папку для загрузок. В зависимости от браузера и настроек устройства, может потребоваться подтверждение для сохранения файла.

Восстановление PLC

Ввести в области Восстановление PLC страницы пароль пользователя для веб-сервера и нажать кнопку "Просмотр" или "Выбрать файл" (в зависимости от используемого браузера), чтобы выбрать сохраненный ранее резервный файл. Нажать на кнопку "Загрузить резервную онлайн-копию" и подтвердить запрос на загрузку этого файла в подключенный PLC. Для этой страницы потребуется право пользователя (Страница 1028) "Восстановление CPU". Если CPU находится в рабочем состоянии RUN и должен быть переведен в состояние STOP, дополнительно потребуется и право "Изменение рабочего состояния".

Во время операции восстановления отображается ряд сообщений о прогрессе данного процесса, и потребуется повторно ввести имя пользователя и пароль. После каждого успешного завершенного шага процесса появляются следующие индикаторы завершения и ссылка для перезагрузки страницы:



⚠ ПРЕДУПРЕЖДЕНИЕ

Восстановление резервных копий с неизвестными содержимым

Восстановление резервной копии с неизвестным содержимым может нанести серьезный материальный ущерб или вред здоровью в случае неисправностей или программных ошибок.

Кроме того, при восстановлении резервной копии, для которой в конфигурации устройства CPU не активирован веб-сервер, нельзя получить доступ к CPU через веб-сервер.

Следует убедиться, что в резервную копию включена конфигурация с известным содержимым.

Примечание

Восстановление резервной копии с иным IP-адресом CPU

При попытке восстановить резервную копию с IP-адресом CPU в резервной копии, отличным от IP-адреса текущего CPU, веб-сервер не сможет отобразить сообщение о завершении восстановления. Если сообщение "Восстановление CPU" остается на экране в течение более пяти минут, то следует ввести новый IP-адрес, соответствующий адресу в файле резервной копии. CPU теперь имеет этот адрес, и можно возобновить доступ к веб-серверу.

12.7.13 Страница архивов данных

На странице "Архивы данных" можно работать с архивами данных из программы STEP 7. Страница позволяет:

- Отобразить список всех архивов данных на PLC
- Загрузить архив данных из PLC на компьютер
- Удалить архив данных из PLC
- Вызвать и очистить архив данных

Архивы данных отображаются по алфавиту в порядке возрастания, без учета регистра. Список архивов данных разбивается на страницы с 50 архивами данных на каждой странице.



Примечание

Управление архивами данных

В файловой системе следует хранить не более 1000 архивов данных. При превышении данного значения веб-серверу может не хватить ресурсов CPU для отображения архивов данных.

Если случится, что веб-страница браузера файлов не сможет отобразить архивы данных, то необходимо перевести CPU в режим STOP, чтобы можно было просматривать и удалять архивы данных.

Управление архивами данных должно быть организовано таким образом, чтобы сохранялось только необходимо число архивов данных и чтобы их количество не превышало максимально допустимого значения, равного 1000 записей.

Работа с архивом данных в Excel

Файл архива данных доступен в американском/английском формате CSV (разделенные запятой значения). Для открытия такого формата в Excel на не американских/английских системах, необходимо импортировать файл с определенными настройками в Excel.

Активное состояние

В столбце "Активный" на странице архивов данных отображается "Да", если для соответствующего файла в CPU есть блок управления архивом данных. Отображается "Нет" при отсутствии такого блока. Можно установить активное состояние на "Да", даже если программа STEP 7 в данный момент не взаимодействует с этим файлом.

Если в программе STEP 7 открыт архив данных или выполняется запись в архив данных, веб-сервер не сможет удалить, загрузить или вызвать и очистить файл архива данных. Кроме того, пока веб-сервер выполняет загрузку архива данных с помощью команды загрузки или команды вызова и очистки, нельзя выполнять никаких других операции с архивом данных, пока загрузка не будет завершена или отменена. Веб-сервер отображает сообщение об ошибке "Приложение занято".

Загрузка файла архивов данных

Можно загрузить файл архива данных, кликнув по его имени. Веб-сервер показывает сообщение об ошибке, если файл отсутствует или выполняется другой процесс загрузки. Сообщение об ошибке отображается на странице до перезагрузки страницы архивов данных. Следующие процессы инициируют перезагрузку страницы архивов данных со стороны веб-сервера:

- Обновление страницы архивов данных или выход со страницы архивов данных и возвращение на страницу
- Изменение установки для количества отображаемых на странице архивов данных
- Успешное удаление архива данных
- Успешный вызов и очистка архива данных

Примечание

Сообщение об ошибке для архивов данных

Автоматическое обновление веб-сервера не удаляет сообщения об ошибках со страницы.

Следующие сценарии вызывают сообщения об ошибках на странице архивов данных:

Процесс	Условие возникновения ошибки	Сообщение об ошибке
Загрузка Вызов/очистка	• Файл отсутствует • Недействительное имя файла • Недопустимый метод HTTP запроса • Карта SMC имеет защиту от записи (только вызов/очистка)	Ошибка при загрузке файла
Удаление	• Недопустимый метод HTTP запроса • Недействительное имя файла • Файл отсутствует	Ошибка при удалении файла
Удаление	• Карта памяти SMC имеет защиту от записи	Ошибка при удалении файла: Карта памяти имеет защиту от записи
Загрузка Удаление Вызов/очистка	• Необходимые права отсутствуют • Метод HTTP запроса не POST или GET • Недействительный ACTION параметр в URL	Файловая операция не разрешена
Загрузка Удаление Вызов/очистка	• Недействительное или отсутствующее поле ссылающегося домена HTTP	Файловая операция не разрешена: нет ссылающегося домена
Загрузка Удаление Вызов/очистка	• Файл архива данных открыт в программе пользователя • Выполняется загрузка архива данных	Приложение занято
Загрузка Удаление Вызов/очистка	• Неожиданная внутренняя ошибка PLC	Внутренняя ошибка

Удаление архива данных

Можно у[illegible]ть архив данных, кликнув в столбце "Удаление" архива данных по символу . Для удаления файла архива данных, подтвердить процесс удаления в диалоговом окне "Удаление".

Вызов и очистка архива данных

Для откр[illegible]тия архива данных и удаления всех записей, кликнуть по символу Вызов и очистка . Для вызова файла архива данных и очистки его содержимое, подтвердить процесс в диалоговом окне Вызов и очистка.

После подтверждения процесса, веб-сервер позволяет загрузить содержимое файла архива данных. В диалоговом окне "Сохранение файла" можно выбрать, будет ли архив данных сохранен или нет. После выбора веб-сервер удаляет содержимое файла архива данных без удаления самого файла. Нельзя отменить процесс очистки архива данных в диалоговом окне "Сохранение файла". Процесс может быть отменен только через начальное подтверждение в диалоговом окне "Вызов и очистка".

Если в процессе вызова и удаления происходит ошибка, то данные из архива не стираются и отображается сообщение об ошибке. Если в столбце "Активный" для архива данных отображается "Нет", то операция вызова и очистки не удаляет содержимое, так

как у программы STEP 7 нет ссылки на файл. В этом случае необходимо удалить файл вручную, чтобы очистить данные.

См. также

Веб-сервер (Страница 1023)

12.7.14 Файлы пользователя

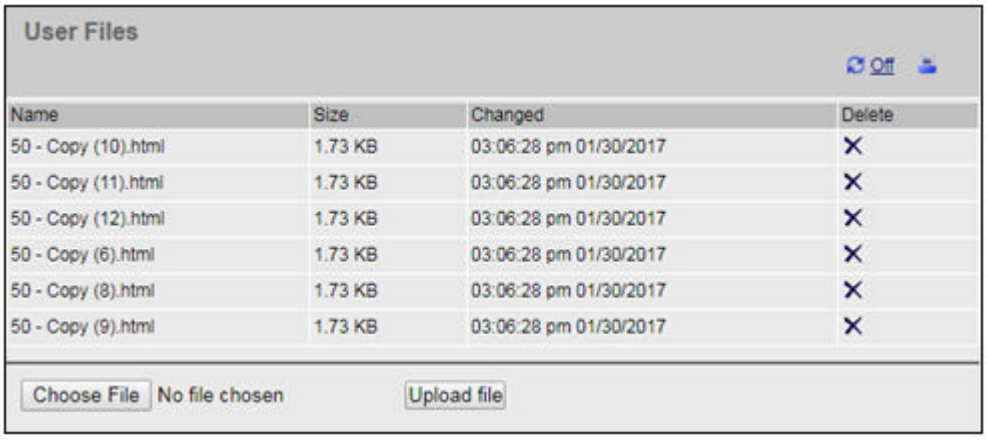
Работа с файлами пользователя

На странице "Файлы пользователя" возможен доступ к файлам во внутренней загружаемой памяти CPU или на карте памяти (внешняя загружаемая память).

Тип доступа к файлам пользователя зависит от конкретных прав пользователя. Все пользователи с правами на чтение файлов могут просматривать файлы в папке на странице "Файлы пользователя". Дополнительно при наличии прав на чтение и запись файлов можно:

- Отобразить список всех файлов пользователя на PLC
- Загрузить файл пользователя из PLC на компьютер
- Выгрузить файл пользователя из компьютера на PLC
- Удалить файл пользователя из PLC

Для просмотра файлов пользователя, нажать на "Файлы пользователя" на главной странице навигации.



Список файлов включает в себя текущий размер файла и дату последнего изменения. Список файлов создается на основе директории "Файлы пользователя " в корневой директории карты SMC.

Примечание

Управление файлами пользователя

На карте SMC следует хранить не более 1000 файлов пользователя. При наличии большего числа файлов, в веб-сервере могут быть отображены только первые 1000 файлов. Файлы отображаются по алфавиту в порядке возрастания, без учета регистра.

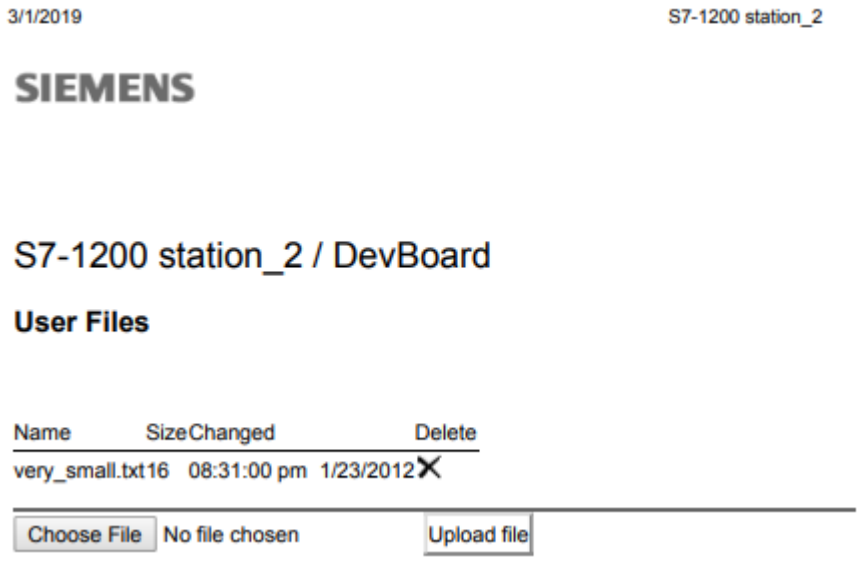
Разбивка списка файлов пользователя на страницы

В списке файлов пользователя на каждой странице отображается по 50 записей. В выпадающем списке можно выбрать диапазон для отображения.



Печать списка файлов пользователя

Можно распечатать список файлов пользователя на PLC, нажав на символ "Печать" на веб-странице PLC:



Ошибки, связанные с файлами пользователя

На странице файлов пользователя отображается сообщение об ошибке, если операция не была успешно завершена:



Сообщение об ошибке отображается на странице до перезагрузки страницы файлов пользователя. Следующие процессы инициируют перезагрузку страницы файлов пользователя со стороны веб-сервера:

- Обновление страницы файлов пользователя или выход со страницы файлов пользователя и возвращение на страницу
- Изменение установки для количества отображаемых на странице файлов пользователя

- Успешное удаление файла пользователя
- Успешная выгрузка файла пользователя

Примечание

Сообщения об ошибках, связанные с файлами пользователя

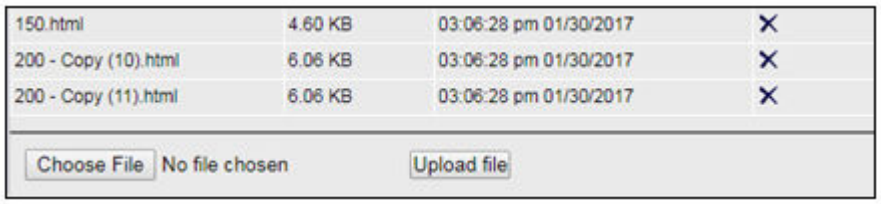
Автоматическое обновление веб-сервера не удаляет сообщения об ошибках со страницы "Файлы пользователя".

Следующие сценарии вызывают сообщения об ошибках на странице файлов пользователя:

Процесс	Условие возникновения ошибки	Сообщение об ошибке
Загрузка файла пользователя	• Файл отсутствует • Внутренняя ошибка PLC	Ошибка при загрузке файла
Загрузка файла пользователя Выгрузка файла пользователя Удаление файла пользователя	• Недостаточно прав для выполнения данной операции • Недопустимый метод HTTP запроса или параметр операции	Файловая операция не разрешена
Загрузка файла пользователя Выгрузка файла пользователя Удаление файла пользователя	• Недействительное или отсутствующее поле ссылающегося домена HTTP	Файловая операция не разрешена - нет ссылающегося домена
Удаление файла пользователя	• Карта памяти имеет защиту от записи	Файловая операция не разрешена
Загрузка Удаление Вызов/очистка	• Файл отсутствует • Внутренняя ошибка PLC	Ошибка при удалении файла - карта памяти с защитой от записи
Выгрузка файла пользователя	• Имя файла отсутствует или недействительно • Внутренняя ошибка PLC	Ошибка при выгрузке файла
Выгрузка файла пользователя	• Имя файла уже существует на PLC	Ошибка при выгрузке файла - имя уже существует
Выгрузка файла пользователя	• Карта SMC заполнена	Ошибка при выгрузке файла - карта памяти заполнена
Выгрузка файла пользователя	• Недействительное имя файла	Ошибка при выгрузке файла - недопустимый символ в имени файла
Выгрузка файла пользователя	• Слишком большой размер файла для файловой системы SM	Ошибка при выгрузке файла - слишком большой файл
Выгрузка файла пользователя	• Карта памяти SMC имеет защиту от записи	Ошибка при выгрузке файла - карта памяти с защитой от записи

Выгрузка файла пользователя

Можно выгрузить файл пользователя, выбрав файл в форме для выгрузки файлов пользователя:



Для выгрузки файла пользователя выбрать файл и нажать на "Выгрузить файл" для запуска передачи файла с компьютера на PLC. Перед началом передачи должны быть выполнены следующие условия:

Необходимое условие	Сообщение
Файл должен быть выбран	Просьба ввести имя файла.
Имя файла должно быть действительным	Допустимые символы - это буквы от "a" до "z", цифры от 0 до 9, знаки (){}[]$!=~ (пробел)
Файл должен быть меньше 2 ГБ (ограничение файловой системы).	Ошибка при выгрузке файла - слишком большой файл

См. также

Веб-сервер (Страница 1023)

12.7.15 API для архивов данных и файлов пользователя

API для архивов данных и файлов пользователя

Для S7-1200 предлагается функция API для архивов данных и файлов пользователя. Более подробная информация содержится в *Справочнике по веб-серверу для S7-1500, ET200SP, ET200pro*..

12.7.16 Браузер файлов

На странице "Файлы пользователя" возможен доступ к файлам во внутренней загружаемой памяти CPU или на карте памяти (внешняя загружаемая память). Страница браузера файлов сначала отображает корневую папку загружаемой памяти, в которой находится папка "Recipes", но при использовании карты памяти отображаются и все другие созданные пользователем папки.

Тип доступа к файлам и папкам зависит от конкретных прав (Страница 1028) пользователя. Все пользователи с правами на чтение файлов могут просматривать файлы в папке в браузере файлов. Независимо от прав пользователя, папка "Recipes" не

может быть удалена. Но если пользовательские папки были созданы на карте памяти, эти папки могут быть удалены после входа в систему с правами на запись и чтение файлов.

Кликнуть по папке для доступа к отдельным файлам в ней.



Рецептурные файлы

В папке рецептур отображаются все рецептурные файлы из загружаемой памяти. Рецептурные файлы также имеют формат CSV и могут быть открыты в Microsoft Excel или другой программе. Для удаления, изменения и сохранения, переименования или выгрузки рецептурных файлов потребуются права на редактирование.

Выгрузка файлов и автоматическое обновление страниц

После начала выгрузки файла, операция выгрузки продолжается до тех пор, пока не будет закрыта веб-страница браузера файлов. Если было активировано автоматическое обновление страниц веб-сервера каждые десять секунд, то каждый раз при обновлении страницы можно увидеть прогресс выгрузки файла. Например, при выгрузке файла размером 2 Мб, обновления размера файла могут отображаться на этапах в 2500, 5000, 10000, 15000, и 20000 байт во время процесса выгрузки.

Если закрыть страницу браузера файлов до завершения процесса выгрузки, то веб-сервер удалит неполный файл.

Дополнительная информация

Примечание

Правила создания имен файлов

Для того, чтобы веб-сервер смог работать с файлами архивов данных и рецептурными файлами, следует использовать символы набора символов ASCII. Символы \ / : * ? " < > | и пробел запрещены.

Если файлы не отвечают этим правилам создания имен, то возможны ошибки веб-сервера при таких операциях, как выгрузка, удаление или переименование файлов. Для переименования файлов во внешней загружаемой памяти может потребоваться картридер и проводник Windows.

Дополнительную информацию о программировании с помощью инструкций архивов данных и о импорте (Страница 506) и экспорте (Страница 504) рецептур можно найти в главе Рецептуры и архивы данных (Страница 499).

См. также

Импорт архивов данных в формате CSV в не американские/английские версии Microsoft Excel (Страница 1114)

12.8 Пользовательские веб-страницы

Для веб-сервера S7-1200 могут создаваться и прикладные пользовательские HTML страницы с данными целевой системы.

⚠ ПРЕДУПРЕЖДЕНИЕ

Несанкционированный доступ к CPU через пользовательские веб-страницы

Несанкционированный доступ к CPU через пользовательские веб-страницы может нарушить ход процесса, что в свою очередь может стать причиной смерти, тяжких телесных повреждений и/или материального ущерба.

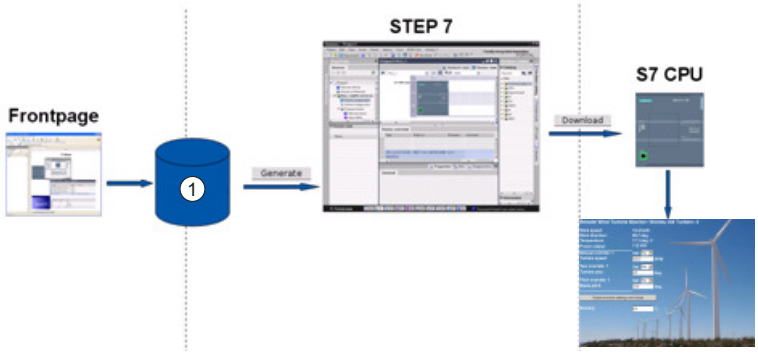
Небезопасное кодирование пользовательских веб-страниц означает уязвимости с точки зрения безопасности, такие как межсайтовый скриптинг (XSS), внедрение кода и другие.

Для защиты S7-1200 CPU от несанкционированного доступа следует ввести S7-1200 CPU в эксплуатацию безопасным образом. Информацию по этой теме можно найти в "Практических рекомендациях" на веб-сайте Промышленной безопасности (http://www.siemens.com/industrialsecurity).

Для создания пользовательских веб-страниц используется HTML-редактор по выбору, готовые страницы загружаются в CPU, где они доступны из меню стандартных веб-страниц. Этот процесс включает в себя несколько задач:

- Создание HTML-страниц с помощью HTML-редактора, напр., Microsoft Frontpage (Страница 1071)
- Включение AWP-команд в HTML-комментарии в HTML-коде (Страница 1072): AWP-команды - это фиксированный набор команд, предоставляемый Siemens для доступа к информации CPU.
- Конфигурирование STEP 7 для чтения и обработки HTML-страниц (Страница 1087)
- Создание блоков из HTML-страниц (Страница 1087)
- Программирование STEP 7 для управления использованием HTML-страниц (Страница 1089)
- Компиляция и загрузка блоков в CPU (Страница 1091)
- Доступ к пользовательским веб-страницам через PC (Страница 1092)

Этот процесс представлен ниже:



① HTML-файлы со встроенными AWP-командами

См. также

Веб-сервер (Страница 1023)

12.8.1 Создание HTML-страниц

Для создания собственных HTML-страниц для веб-сервера можно использовать пакет программного обеспечения по выбору. Следует убедиться, что HTML-код соответствует HTML-стандартам W3C (консорциум World Wide Web). STEP 7 не выполняет проверку HTML-синтаксиса.

Можно использовать ПО для программирования в режиме WYSIWYG или дизайн-макета, но необходима и возможность для редактирования своего кода HTML только в формате HTML. Большинство приложений для создания веб-страниц обеспечивают этот тип редактирования. В противном случае, всегда можно использовать простой текстовый

редактор, чтобы отредактировать HTML-код. Вставить следующую строку в свою HTML-страницу, чтобы выбрать для страницы набор символов UTF-8:
<meta http-equiv="content-type" content="text/html; charset=utf-8">

Также следует убедиться, что файл сохраняется в редакторе также в кодировке UTF-8:

Можно использовать STEP 7, чтобы скомпилировать все содержимое HTML-страниц в блоки данных STEP 7. Эти блоки данных состоят из одного управляющего блока, который отвечает за отображение веб-страниц, и одного или нескольких фрагментных блоков данных, которые содержат скомпилированные веб-страницы. Следует учитывать, что HTML-страницы с большим объемом информации, в первую очередь графической, занимают много места в загружаемой памяти (Страница 1092) для DB фрагментов. Если внутренней загружаемой памяти CPU недостаточно для пользовательских веб-страниц, то следует использовать карту памяти (Страница 136) в качестве внешней загружаемой памяти.

Для программирования HTML-кода таким образом, чтобы можно было использовать данные из S7-1200, можно вставлять AWP-команды (Страница 1072) как HTML-комментарии. После следует сохранить HTML-станицы на PC и запомнить путь сохранения.

Примечание

Размер файла HTML с командами AWP не может превышать 64 кБ. Для правильной компиляции страниц в STEP 7, размер файлов не должен превышать данного ограничения.

Обновление пользовательских веб-страниц

Пользовательские веб-страницы не обновляются автоматически. Можно выбрать программирование HTML кода с обновление страницы или без такового. Для страниц с данными PLC регулярное обновление позволяет всегда поддерживать эти данные в актуальном состоянии. Для HTML-страниц, представляющих собой формы для заполнения, обновление может помешать вводу данных пользователем. Если вся страница должна обновляться автоматически, то можно вставить следующую строку в заголовок HTML, где "10" означает число секунд между двумя процессами обновления:
<meta http-equiv="Refresh" content="10">

Для обновления страниц и данных также можно использовать JavaScripts или другие HTML-средства. Описание можно найти в документации по HTML или JavaScript.

12.8.2 Поддерживаемые веб-сервером S7-1200 команды AWP

Веб-сервер S7-1200 предлагает AWP-команды, которые могут встраиваться в пользовательские веб-страницы как HTML-комментарии в следующих целях:

- Чтение переменных (Страница 1074)
- Запись переменных (Страница 1075)
- Чтение специальных переменных (Страница 1077)

- Запись специальных переменных (Страница 1079)
- Определение типов перечисления (Страница 1081)
- Назначение переменных типам перечисления (Страница 1082)
- Создание фрагментных блоков данных (Страница 1083)

Общий синтаксис

За исключением команды чтения переменной, AWP-команды имеют следующий синтаксис:

```
<!-- AWP_ <command name and parameters> -->
```

AWP-команды в сочетании с типовыми командами HTML-форм используются для записи в переменные в CPU.

Описания AWP-команд на следующих страницах используют следующие правила:

- Элементы, заключенные в квадратные скобки [], являются необязательными.
- Элементы, заключенные в угловые скобки < >, являются значениями параметров, которые должны быть определены.
- Кавычки обозначают литеральную часть команды. Они должны присутствовать, как обозначено.
- Специальные символы в именах переменных (тегов) или блоков данных, в зависимости от использования, обозначаются через символьную строку ESC или заключаются в кавычки (Страница 1085).

С помощью текстового редактора или режима HTML-редактирования, AWP-команды вставляются на страницах.

Примечание

Ожидаемый синтаксис AWP-команд

Пробел после "<!--" и пробел перед "-->"в определении AWP-команды важны для надлежащей компиляции команды. Пропуск пробелов может помешать компилятору сгенерировать надлежащий код. Компилятор в этом случае не отображает ошибку.

Обзор AWP-команд

Детальные сведения по использованию каждой из AWP-команд содержатся в следующих разделах, здесь же представлен краткий обзор команд:

Чтение переменных

```
:=<Varname>:
```

Запись переменных

```
<!-- AWP_In_Variable Name='<Varname1>' [Use='<Varname2>'] ... -->
```

Эта AWP-команда просто объявляет, что возможна запись в переменную спецификатора имени. HTML-код выполняет запись в переменную по имени из <input>, <select> или других HTML-операторов в HTML-форме.

Чтение специальных переменных
```
<!-- AWP_Out_Variable Name='<Type>:<Name>' [Use='<Varname>'] -->
```

Запись специальных переменных
```
<!-- AWP_In_Variable Name='<Type>:<Name>' [Use='<Varname>']-->
```

Определение типов перечисления
```
<!-- AWP_Enum_Def Name='<Enum type
name>' Values='<Value>, <Value>,... ' -->
```

Определение типов перечисления
```
<!-- AWP_In_Variable Name='<Varname>' Enum="<Enum type name>" -->
<!-- AWP_Out_Variable Name='<Varname>' Enum="<Enum type name>" -->
```

Создание фрагментов
```
<!-- AWP_Start_Fragment Name='<Name>' [Type=<Type>][ID=<ID>] -->
```

Импорт фрагментов
```
<!-- AWP_Import_Fragment Name='<Name>' -->
```

12.8.2.1 Чтение переменных

Пользовательские веб-страницы могут читать переменные (переменные (теги) PLC) и переменные (теги) блока данных из CPU при условии, что был сконфигурирован доступ к переменным (тегам) из HMI.

Синтаксис

```
:=<Varname>:
```

Параметр

<Varname>	Переменная, которую необходимо прочитать, которая может быть именем переменной (тега) PLC из программы STEP 7, переменной (тегом) блока данных, каналом ввода-вывода или адресом в памяти. Для адресов в памяти или адресов ввода-вывода или псевдонимов (Страница 1085) не следует заключать в одиночные кавычки имя переменной (тега). Имена переменных (тегов) PLC заключаются в двойные кавычки. Для переменных (тегов) блока данных только имя блока заключается в двойные кавычки. Имя переменной (тега) находится вне кавычек. Следует обратить внимание на то, что используется имя, а не номер блока данных. Для ссылок на элементы массива следует использовать синтаксис для элементов массива.

Примеры

```
:="Conveyor_speed":
:="My_Data_Block".flag1:
:=I0.0:
:=MW100:
:="My_Data_Block".Array_Dim1[0]:
:="My_Data_Block".Array_Dim2[0,0]:
```

Пример чтения переменных-псевдонимов

```
<!-- AWP_Out_Variable Name='flag1' Use='"My_Data_Block".flag1' -->
:=flag1:
```

Примечание

Определение псевдонимов для переменных (тегов) PLC и PLC блоков данных описано в теме Использование псевдонимов для ссылки на переменные (Страница 1080).

Если имя переменной (тега) или имя блока данных содержит специальные символы, то следует использовать дополнительные кавычки или символы ESC. Описание см. в Обработка имен переменных (тегов), содержащих специальные символы (Страница 1085).

12.8.2.2 Запись переменных

Пользовательские веб-страницы могут записывать данные в CPU. Для этого с помощью AWP-команды в CPU указывается переменная для записи через HTML-страницу. Переменная должна быть указана по имени переменной (тега) PLC или по имени переменной (тега) блока данных. Можно объявить несколько имен переменных в одном операторе. Для записи данных в CPU используется стандартная HTTP команда POST.

Типичным случаем использования является разработка формы на HTML-странице с полями ввода текста или списками выбора, которые соответствуют перезаписываемым переменным CPU. Как и в случае других пользовательских веб-страниц, затем генерируются блоки STEP 7 для включения в программу STEP 7. Когда пользователь с привилегиями по изменению переменных впоследствии получает доступ к этой странице и вводит данные в поля ввода или выбирает значение из списка, веб-сервер преобразовывает введенные значения в надлежащий тип данных для переменной и записывает значение в переменную в CPU. Следует учитывать, что спецификатор имени для HTML полей ввода и HTML списков выбора использует синтаксис, типичный для спецификаторов имени команды AWP_In_Variable . Имя обычно заключается в одиночные кавычки, а при ссылке на блок данных, имя блока данных заключается в двойные кавычки.

Для получения дополнительной информации по управлению формами обратиться к документации по HTML.

Синтаксис

```
<!-- AWP_In_Variable Name='<Varname1>' [Use='<Varname2>'] ... -->
```

Параметр

<Varname1>	Если не указан спецификатор использования, то Varname1 является переменной, в которую будет выполнена запись. Это может быть имя переменной (тега) PLC из программы STEP 7 или переменная (тег) из определенного блока данных или имя блока данных. Если спецификатор использования указан, то Varname1 является альтернативным именем для переменной, упомянутой в <Varname2> (Страница 1080). Это локальное имя на HTML-странице.
<Varname2>	Если не указан спецификатор использования, то Varname2 является переменной, в которую будет выполнена запись. Это может быть имя переменной (тега) PLC из программы STEP 7 или переменная (тег) из определенного блока данных.

Как для элемента имени, так и для спецификатора использования, полное имя должно быть заключено в одиночные кавычки. Внутри одиночных кавычек двойные кавычки используются для выделения переменной (тега) PLC и имени блока данных. В двойные кавычки заключается имя блока данных, но не имя переменной (тега) блока данных. Следует помнить, что для переменной (тега) блока данных используется имя блока, а не номер блока данных. Для ссылок на элементы массива следует использовать синтаксис для элементов массива.

Если с помощью команды AWP_In_Variable блок данных разрешается для записи, то запись возможна для любой переменной (тега) в блоке данных.

Примеры с полями ввода HTML

```
<!-- AWP_In_Variable Name='"Target_Level"' -->
<form method="post">
<p>Input Target Level: <input name='"Target_Level"' type="text" />
</p>
</form>

<!-- AWP_In_Variable Name='"Data_block_1".Braking' -->
<form method="post">
<p>Braking: <input name='"Data_block_1".Braking' type="text" /> %</p>
</form>

<!-- AWP_In_Variable Name='"Data_block_1".Array_Dim2' -->
<form method="post">
<p>Two-dimensional array value [2,1]: <input
name='"Data_block_1".Array_Dim2[2,1]' type="text" /> %</p>
</form>
```

Пример для спецификатора использования

```
<!-- AWP_In_Variable Name='"Braking"' Use='"Data_block_1".Braking' --
>
<form method="post">
<p>Braking: <input name='"Braking"' type="text" /> %</p>
</form>
```

Пример с записываемым блоком данных

```
<!-- AWP_In_Variable Name='"Data_block_1"' -->
<form method="post">
<p>Braking: <input name='"Data_block_1".Braking' type="text" /> %
</p>
<p>TurbineSpeed: <input name='"Data_block_1".TurbineSpeed' size="10"
value='"Data_block_1".TurbineSpeed' type="text" />
</p>
</form>
```

Пример со списком выбора HTML

```
<!-- AWP_In_Variable Name='"Data_block_1".ManualOverrideEnable'-->
<form method="post">
<select name='"Data_block_1".ManualOverrideEnable'>
<option value=:"Data_block_1".ManualOverrideEnable:> </option>
<option value=1>Yes</option>
<option value=0>No</option>
</select><input type="submit" value="Submit setting" /></form>
```

Примечание

Только пользователь с правом (Страница 1028) записи на пользовательские веб-страницы может записывать данные в CPU. Веб-сервер игнорирует команды, если у пользователя нет прав редактирования.

Если имя переменной (тега) или имя блока данных содержит специальные символы, то следует использовать дополнительные кавычки или символы ESC. Описание см. в "Обработка имен переменных (тегов), содержащих специальные символы (Страница 1085)".

12.8.2.3 Чтение специальных переменных

Веб-сервер предлагает возможность по считыванию значений из устройства PLC для сохранения их в специальных переменных в HTTP-заголовке ответа. Можно, например, прочитать имя пути из переменной (тега) PLC, чтобы перенаправить URL в другое расположение, используя специальную переменную HEADER:Location.

Синтаксис

```
<!-- AWP_Out_Variable Name='<Type>:<Name>' [Use='<Varname>'] -->
```

Параметры

<Typ>	Один из следующих типов специальной переменной: HEADER COOKIE_VALUE COOKIE_EXPIRES
<Name>	В документации по HTTP можно найти список всех имен переменных HEADER. Несколько примеров приведены ниже: Status: Код ответа Location: Путь для перенаправления Retry-After: Ожидаемое время недоступности службы для запрашивающего клиента. Для типов COOKIE_VALUE и COOKIE_EXPIRES, <Name> является именем определенного "куки". COOKIE_VALUE:name: Значение названного "куки" COOKIE_EXPIRES:name: Время окончания срока действия названного "куки" в секундах Спецификатор имени должен быть указан в простых или двойных кавычках. Если спецификатор использования указан, то имя специальной переменной соответствует имени переменной (тега) PLC. Заключить весь спецификатор имени в одиночные кавычки, а переменную (тег) PLC в двойные кавычки. Имя специальной переменной и имя переменной (тега) PLC должны совпадать.
<Varname>	Имя переменной (тега) PLC или переменной (тега) блока данных для считывания переменной. Varname должно быть заключено в одиночные кавычки. Внутри одиночных кавычек двойные кавычки используются для выделения переменной (тега) PLC и имени блока данных. В двойные кавычки заключается имя блока данных, но не имя переменной (тега) блока данных. Следует помнить, что для переменной (тега) блока данных используется имя блока, а не номер блока данных.

Если имя переменной (тега) или имя блока данных содержит специальные символы, то следует использовать дополнительные кавычки или символы ESC. Описание см. в Обработка имен переменных (тегов), содержащих специальные символы (Страница 1085).

Пример: Чтение специальной переменной без спецификатора использования

```
<!-- AWP_Out_Variable Name='"HEADER:Status"' -->
```

В этом примере специальные переменная HTTP "HEADER:Status" принимает значения переменной (тега) PLC "HEADER:Status". Имя в таблице переменных (тегов) PLC должно точно соответствовать имени специальной переменной, если спецификатор использования не указан.

Пример: Чтение специальной переменной со спецификатором использования

```
<!-- AWP_Out_Variable Name='HEADER:Status' Use='"Status"' -->
```

В этом примере специальная переменная "HEADER:Status" принимает значения переменной (тега) PLC "Status".

12.8.2.4 Запись специальных переменных

Веб-сервер предлагает возможность для записи значений из специальных переменных в заголовке HTTP-запроса в CPU. Например, в STEP 7 можно сохранять информацию о куки пользовательской веб-страницы, пользователе, который обращается к странице, или данные заголовка. Веб-сервер обеспечивает доступ к определенным специальным переменным, которые могут записываться в CPU пользователем, авторизованным с правами по изменению переменных.

Синтаксис

```
<!-- AWP_In_Variable Name='<Type>:<Name>' [Use='<Varname>']-->
```

Параметры

<Typ>	Тип специальной переменной. Возможно: HEADER SERVER COOKIE_VALUE
<Name>	Специальная переменная одного из определенных выше типов, как показано в данных примерах: HEADER:Accept: Разрешенные типы содержимого HEADER:User-Agent: Информация о пользовательском агенте, от которого поступает запрос. SERVER:current_user_id: ID текущего пользователя; 0 при отсутствии зарегистрированного пользователя SERVER:current_user_name: Имя текущего пользователя COOKIE_VALUE:<name>: Значение названного "куки" Заключить спецификатор имени в одиночные кавычки. Если спецификатор использования указан, то имя специальной переменной соответствует имени переменной (тега) PLC. Заключить весь спецификатор имени в одиночные кавычки, а переменную (тег) PLC в двойные кавычки. Имя специальной переменной должно точно соответствовать имени переменной (тега) PLC. В документации по HTTP можно найти список всех имен переменных HEADER.
<Varname>	Имя переменной в программе STEP 7, в которую необходимо записать специальную переменную. Речь может идти о переменной (теге) PLC или переменной (теге) блока данных. Varname должно быть заключено в одиночные кавычки. Внутри одиночных кавычек двойные кавычки используются для выделения переменной (тега) PLC и имени блока данных. В двойные кавычки заключается имя блока данных, но не имя переменной (тега) блока данных. Следует помнить, что для переменной (тега) блока данных используется имя блока, а не номер блока данных.

Примеры

```
<!-- AWP_In_Variable Name='"SERVER:current_user_id"' -->
```

В этом примере веб-страница записывает значение специальной HTTP-переменной "SERVER:current_user_id" в переменную (тег) PLC "SERVER:current_user_id".

```
<!-- AWP_In_Variable Name=SERVER:current_user_id' Use='"my_userid"'
-->
```

В этом примере веб-страница записывает значение специальной HTTP-переменной "SERVER:current_user_id" в переменную (тег) PLC "my_userid".

Примечание

Только пользователь с правом на редактирование переменных может записывать данные в CPU. Веб-сервер игнорирует команды, если у пользователя нет прав редактирования.

Если имя переменной (тега) или имя блока данных содержит специальные символы, то следует использовать дополнительные кавычки или символы ESC. Описание см. в "Обработка имен переменных (тегов), содержащих специальные символы (Страница 1085)".

12.8.2.5 Использование псевдонимов для ссылки на переменные

Для In_Variable или Out_Variable можно использовать псевдоним на своей пользовательской веб-странице. Например, можно использовать на своей HTML-странице символьное имя, отличное от того, которое используется в CPU, либо можно приравнять переменную в CPU к специальной переменной. Спецификатор использования AWP предлагает такую возможность.

Синтаксис

```
<-- AWP_In_Variable Name='<Varname1>' Use='<Varname2>' -->
<-- AWP_Out_Variable Name='<Varname1>' Use='<Varname2>' -->
```

Параметры

<Varname1>	Псевдоним или имя специальной переменной Varname1 должно быть заключено в одиночные или двойные кавычки.
<Varname2>	Имя переменной PLC, которой необходимо присвоить псевдоним. Речь может идти о переменной (теге) PLC или переменной (теге) блока данных или специальной переменной. Varname2 должно быть заключено в одиночные кавычки. Внутри одиночных кавычек двойные кавычки используются для выделения переменной (тега) PLC, специальной переменной или имени блока данных. В двойные кавычки заключается имя блока данных, но не имя переменной (тега) блока данных. Следует помнить, что для переменной (тега) блока данных используется имя блока, а не номер блока данных.

Примеры

```
<-- AWP_In_Variable Name='SERVER:current_user_id'
Use='"Data_Block_10".server_user' -->
```

В этом примере специальная переменная SERVER:current_user_id записывается в переменную (тег) "server_user" в блоке данных "Data_Block_10".

```
<-- AWP_Out_Variable Name='Weight'
Use='"Data_Block_10".Tank_data.Weight'  -->
```

В этом примере на значение структурного элемента блока данных Data_Block_10.Tank_data.Weight можно сослаться просто с помощью "Weight" везде в остальной части пользовательской веб-страницы.

```
<-- AWP_Out_Variable Name='Weight' Use='"Raw_Milk_Tank_Weight"'  -->
```

В этом примере на значение переменной (тега) PLC "Raw_Milk_Tank_Weight" можно сослаться просто с помощью "Weight" везде в остальной части пользовательской веб-страницы.

Если имя переменной (тега) или имя блока данных содержит специальные символы, то следует использовать дополнительные кавычки или символы ESC. Описание см. в Обработка имен переменных (тегов), содержащих специальные символы (Страница 1085).

12.8.2.6 Определение типов перечисления

Можно определять типы перечислений на своих пользовательских страницах и назначать элементы в AWP-команде.

Синтаксис

```
<!-- AWP_Enum_Def Name='<Enum type name>' Values='<Value>,
<Value>,... ' -->
```

Параметры

<Enum type name>	Имя типа перечисления, заключенное в одиночные или двойные кавычки.
<Value>	<константа>:<имя> Константа указывает числовое значение для назначения типа перечисления. Общее количество не ограничено. Имя - это значение, назначенное элементу перечисления.

Обратить внимание на то, что вся строка назначенных значений перечисления заключена в одиночные кавычки, а каждое отдельное назначение элемента для типа перечисления заключено в двойные кавычки. Сфера действия определения типа перечисления является глобальной для пользовательских веб-страниц. Если пользовательские веб-страницы были помещены в языковые папки (Страница 1105), то определение типа перечисления является глобальным для всех страниц языковой папки.

Пример

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"No alarms", 1:"Tank is
full", 2:"Tank is empty"' -->
```

12.8.2.7 Ссылка на переменные CPU с типом перечислений

Можно присвоить переменной в CPU тип перечисления. Эта переменная может использоваться в других местах пользовательской веб-страницы в операции чтения (Страница 1074) или операции записи (Страница 1075). В операции чтения веб-сервер заменит числовое значение, которое было считано из CPU, на соответствующее текстовое значение перечисления. В операции записи веб-сервер заменит текстовое значение на целочисленное значение перечисления, которое соответствует тексту, прежде чем записать значение в CPU.

Синтаксис

```
<!-- AWP_In_Variable Name='<Varname>' Enum="<EnumType>" -->
<!-- AWP_Out_Variable Name='<Varname>' Enum="<EnumType>" -->
```

Параметры

<Varname>	Имя переменной (тега) PLC или переменной (тега) блока данных для назначения типу перечисления, или псевдоним для переменной (тега) PLC (Страница 1080), если объявлен. Имя переменной должно быть заключено в одиночные кавычки. Внутри одиночных кавычек двойные кавычки используются для выделения переменной (тега) PLC и имени блока данных. Следует помнить, что для переменной (тега) блока данных используется имя блока, а не номер блока данных. В двойные кавычки заключается имя блока данных, но не имя переменной (тега) блока данных.
<EnumType>	Имя типа перечисления, заключенное в одиночные или двойные кавычки

Адресация типа перечисления действует в текущем фрагменте.

Пример использования при чтении переменной

```
<!-- AWP_Out_Variable Name='"Alarm"' Enum="AlarmEnum" -->...
<p>The current value of "Alarm" is :="Alarm":</p>
```

Если значение "Alarm" в CPU равняется 2, то HTML-страница отображает 'The current value of "Alarm" is Tank is empty', поскольку определение типа перечисления (Страница 1081) назначает текстовую строку "Tank is empty" числовому значению 2.

Пример использования при записи переменной

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"No alarms", 1:"Tank is
full", 2:"Tank is empty"' -->
<!-- AWP_In_Variable Name='"Alarm"' Enum='AlarmEnum' -->...
<form method="POST">
<p><input type="hidden" name='"Alarm"' value="Tank is full" /></p>
<p><input type="submit" value='Set Tank is full' /><p>
</form>
```

Так как определение типа перечисления (Страница 1081) назначает текст "Tank is full" числовому значению 1, значение 1 записывается в переменную (тег) PLC "Alarm" в CPU.

Обратить внимание на то, что спецификатор перечисления в объявлении AWP_In_Variable должен точно соответствовать спецификатору имени в объявлении AWP_Enum_Def .

Примеры для записи переменных с использованием псевдонимов

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"No alarms", 1:"Tank is
full", 2:"Tank is empty"' -->
<!-- AWP_In_Variable Name='"Alarm"' Enum='AlarmEnum'
Use='"Data_block_4".Motor1.Alarm'-->...
<form method="POST">
<p><input type="hidden" name='"Alarm"' value="Tank is full" /></p>
<p><input type="submit" value='Set Tank is full' /><p>
</form>
```

Так как определение типа перечисления (Страница 1081) назначает текст "Tank is full" числовому значению 1, значение 1 записывается в псевдоним "Alarm", соответствующий переменной (тегу) PLC "Motor1.Alarm" в блоке данных "Data_Block_4" CPU.

Если имя переменной (тега) или имя блока данных содержит специальные символы, то следует использовать дополнительные кавычки или символы ESC. Описание см. в Обработка имен переменных (тегов), содержащих специальные символы (Страница 1085).

Примечание

Прежние версии требовали отдельного объявления AWP_Enum_Ref для назначения переменной определенного типа перечисления. STEP 7 и S7-1200 поддерживают существующий код с AWP_Enum_Ref объявлениями. Однако необходимости в этой команде больше нет.

12.8.2.8 Создание фрагментов

При нажатии на кнопку "Создание блоков" в свойствах CPU для веб-сервера, STEP 7 преобразовывает и сохраняет пользовательские веб-страницы как управляющие DB и фрагментные DB. Можно устанавливать определенные фрагменты для определенных страниц или для разделов определенных страниц. Можно обозначить эти фрагменты по имени и номеру с помощью AWP-команды "Start_Fragment". Всё содержимое страницы после команды AWP_Start_Fragment, относится к этому фрагменту, пока не будет использована другая команда AWP_Start_Command или пока не будет достигнут конец файла.

Синтаксис

```
<!-- AWP_Start_Fragment Name='<Name>' [Type=<Typ>][ID=<ID>]
[Mode=<Mode>] -->
```

Параметры

<Name>	Текстовая строка: Имя фрагментного DB Имя фрагмента должно начинаться с буквы или символа подчеркивания и состоять из букв, цифр и подчеркиваний. Имя фрагмента является регулярным выражением формы: [a-zA-Z_][a-zA-Z_0-9]*
<Typ>	Ручной или автоматический режим Ручной режим: Программа STEP 7 должна запросить этот фрагмент и может ответить соответствующим образом. Работой фрагмента нужно управлять с помощью STEP 7 и переменных управляющего DB. Автоматический режим: Веб-сервер обрабатывает фрагмент автоматически. Если параметр для типа не указывается, то по умолчанию используется "Автоматический режим".
<ID>	Целочисленный идентификационный номер. Если параметр ID не указывается, веб-сервер по умолчанию назначает число. Для ручных фрагментов, установить для ID небольшое число. ID является средством, с помощью которого программа STEP 7 управляет ручным фрагментом.
<Mode>	"Видимое" или "Скрытое" Видимое: Содержимое фрагмента отображается на пользовательской веб-странице. Скрытое: Содержимое фрагмента не отображается на пользовательской веб-странице. Если параметр для типа не указывается, то по умолчанию используется "Видимое".

Ручные фрагменты

При создании ручного фрагмента для пользовательской веб-страницы или части страницы, программа STEP 7 должна выбирать момент отправки фрагмента. Программа STEP 7 должна установить надлежащие параметры в управляющем DB для пользовательской страницы при ручном управлении и затем вызвать инструкцию WWW с измененным управляющим DB. Для понимания структуры управляющего DB и того, как изменять отдельные страницы и фрагменты, см. "Расширенное управление пользовательскими веб-страницами (Страница 1108)".

12.8.2.9 Импорт фрагментов

Можно создать именованный фрагмент из части HTML-кода и затем импортировать этот фрагмент в другое место в наборе пользовательских веб-страниц. Например, представим ряд пользовательских веб-страниц, у которого есть начальная страница, а также несколько других HTML-страниц, доступных с помощью ссылок на начальной странице. Предположим, что каждая из отдельных страниц должна отображать логотип компании. Для этого можно создать фрагмент (Страница 1083), который загружает изображение логотипа компании. Каждая отдельная HTML-страница могла бы в этом случае импортировать этот фрагмент, чтобы отобразить логотип компании. С этой целью используется AWP команда Import_Fragment. HTML-код для фрагмента существует только в одном фрагменте, но можно импортировать этот фрагментный DB столько раз, сколько необходимо на веб-страницах по выбору.

Синтаксис

```
<!-- AWP_Import_Fragment Name='<Name>' -->
```

Параметры

<Name>	Текстовая строка: Имя импортируемого фрагментного DB.

Пример

Часть HTML-кода, которая создает фрагмент для отображения изображения:

```
<!-- AWP_Start_Fragment Name='My_company_logo' --><p><img
src="company_logo.jpg"></p>
```

Часть HTML-кода в другом файле .html, импортирующем фрагмент с изображение логотипа:

```
<!-- AWP_Import_Fragment Name='My_company_logo' -->
```

Оба *.html файла (один, который создает фрагмент и второй, который импортирует его) находятся в структуре папок, которая определяется при конфигурировании пользовательских страниц в STEP 7 (Страница 1087).

12.8.2.10 Объединение определений

При объявлении переменных для использования на пользовательских веб-страницах можно объединить объявление переменной и псевдоним для переменной (Страница 1080). Также можно объявить несколько In_Variables в одном операторе и несколько Out_Variables в одном операторе.

Примеры

```
<!-- AWP_In_Variable Name='"Level"' Name='"Gewicht"' Name='"Temp"' --
>
<!-- AWP_Out_Variable Name='HEADER:Status' Use='"Status"'
         Name='HEADER:Location' Use="Location"
         Name='COOKIE_VALUE:name' Use="my_cookie" -->
<!-- AWP_In_Variable Name='Alarm' Use='"Data_block_10".Alarm' -->
```

12.8.2.11 Обработка имен переменных (тегов), содержащих специальные символы

При указании имен переменных на пользовательских веб-страницах необходимо проявить особенную осторожность, если имена переменных (тегов) содержат символы, имеющие особое значение.

Чтение переменных

Для чтения переменной (Страница 1074) следует использовать следующий синтаксис:

```
:=<Varname>:
```

Следующие правила действуют при чтении переменных:

- Имена переменных из таблицы переменных (тегов) PLC заключаются в двойные кавычки.
- Имена переменных, в случае которых речь идет о переменных (тегах) блоков данных, имя блока данных заключается в двойные кавычки. Переменная (тег) находится вне кавычек.
- Имена переменных, которые являются прямыми адресами ввода-вывода, адресами памяти или псевдонимами, не используют кавычки для читаемой переменной.
- Для имен переменных (тегов) или имен переменных (тегов) блока данных, которые содержат обратную косую черту, перед обратной косой чертой должна быть установлена другая обратная косая черта.
- Если имя переменной (тега) или имя переменной (тега) блока данных содержит двоеточие, знак меньше, знак больше, или амперсанд, то для считываемой переменной должен быть определен псевдоним, у которого нет специальных символов, для считывания переменной через этот псевдоним. Перед двоеточием в именах переменных (тегов) в элементах использования ставится обратная косая черта.

Таблица 12-1 Пример для читаемых переменных

Имя блока данных	**Имя переменной (тега)**	**Команда чтения**
-/-	ABC:DEF	`<!--AWP_Out_Variable Name='special_tag' Use ='"ABC:DEF"' --> :=special_tag:`
Неприменимо	T\	`:="T\\":`
Неприменимо	A \B 'C :D	`<!--AWP_Out_Variable Name='another_special_tag' Use='"A \\B \'C :D"' --> :=another_special_tag:`
Неприменимо	a<b	`<!--AWP_Out_Variable Name='a_less_than_b' Use='"a<b"' --> :=a_less_than_b:`
Data_block_1	Tag_1	`:="Data_block_1".Tag_1:`
Data_block_1	ABC:DEF	`<!-- AWP_Out_Variable Name='special_tag' Use='"Data_block_1".ABC\:DEF'--> :=special_tag:`
DB A' B C D$ E	Tag	`:="DB A' B C D$ E".Tag:`
DB:DB	Tag:Tag	`<!--AWP_Out_Variable Name='my_tag' Use ='"DB:DB".Tag\:Tag' --> :=my_tag:`

Спецификаторы имен и использования

AWP-команды AWP_In_Variable, AWP_Out_Variable, AWP_Enum_Def, AWP_Enum_Ref, AWP_Start_Fragment и AWP_Import_Fragment имеют спецификаторы имен. Такие команды HTML-форм как <input> и <select> также имеют спецификаторы имен. У AWP_In_Variable и AWP_Out_Variable дополнительно могут иметь спецификаторы использования. Независимо от команды, синтаксис для спецификаторов имени и использования в части использования специальных символов один и тот же:

- Текст, указанный для спецификаторов имени или использования, должен быть заключен в одиночные кавычки. Если заключенное в кавычки имя является переменной (тегом) PLC или именем блока данных, следует использовать одиночные кавычки для всего спецификатора.
- Внутри спецификатора имени или использования имена блока данных и переменной (тега) PLC должны быть заключены в двойные кавычки.
- Если имя переменной (тега) или имя блока данных содержит простые кавычки или обратную косую черту ("Backslash"), то следует установить перед таким символом обратную косую черту в качестве символа ESC. Обратная косая черта "\" является символом перехода в компиляторе AWP-команд.

Таблица 12-2 Примеры для спецификаторов имен

Имя блока данных	Имя переменной (тега)	Возможности для спецификаторов имен
-/-	ABC'DEF	`Name='"ABC\'DEF"'`
Неприменимо	A \B 'C :D	`Name='"A \\B \'C :D"'`
Data_block_1	Tag_1	`Name='"Data_block_1".Tag_1'`
Data_block_1	ABC'DEF	`Name='"Data_block_1".ABC\'DEF'`
Data_block_1	A \B 'C :D	`Name='"Data_block_1".A \\B \'C :D'`
DB A' B C D$ E	Tag	`Name='"DB A\' B C D$ E".Tag'`

Правила для спецификаторов использования идентичны правилам для спецификаторов имен.

Примечание

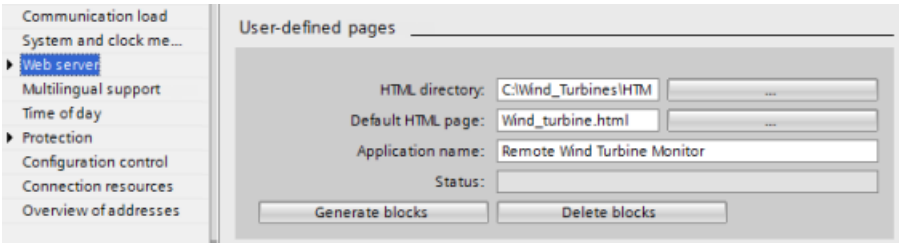
Независимо от того, какие символы используются на HTML-странице, установить для HTML-страницы набор символов UTF-8 и сохранять страницу в редакторе с кодировкой символов UTF-8.

12.8.3 Конфигурирование использования пользовательских веб-страниц

Чтобы сконфигурировать пользовательские веб-страницы в STEP 7, выполнить следующие действия:

1. Выбрать CPU в конфигурации устройства.
2. Вызвать в окне инспектора CPU свойства веб-сервера.

3. Если еще не было сделано, установить флажок для "Активировать веб-сервер на этом модуле".
4. Выбрать "Разрешен только HTTPS доступ" для шифрования коммуникации веб-сервером и защиты доступного по сети CPU.
5. Ввести имя каталог на PC, где была сохранена HTML-страница по умолчанию (начальная страница).
6. Ввести имя страницы по умолчанию.
7. Присвоить имя своему приложению (по желанию). Веб-сервер использует имя приложения, чтобы упорядочи по подкатегориям или сгруппировать веб-страницы. После ввода имени приложения, веб-сервер создает URL для пользовательской страницы в следующем формате: http[s]://ww.xx.yy.zz/awp/<имя приложения>/<имя страницы>.html. Wird kein Anwendungsname eingegeben, lautet die URL без ввода имени приложения имеет следующий формат http[s]://ww.xx.yy.zz/awp/<имя страницы>.html.
   Избегать специальных символов в имени приложения. Некоторые символы могут привести к невозможности для веб-сервера отображать пользовательские страницы.



8. Ввести в разделе "Дополнительно" Введите расширения имен файлов, которые содержат AWP-команды. По умолчанию STEP 7 анализирует файлы с расширениями *.htm, *.html и *.js. При использовании файлов с другими расширениями, добавить их. Для экономии ресурсов процесса, не вводить расширений файлов, если файлы такого типа не содержат AWP команд.
9. Сохранить значение по умолчанию для номера веб-DB или ввести свой номер. Это номер DB управления, который отвечает за отображение веб-страниц.
10. Сохранить стартовый номер по умолчанию для фрагментного DB или ввести свой номер. Это первый из фрагментных DB с веб-страницами.

Генерация программных блоков

Если нажать кнопку "Генерация блоков", STEP 7 генерирует блоки данных из HTML-страниц в указанном исходном HTML-каталоге, и управляющий блок для работы с веб-страницами. Можно установить эти атрибуты по мере необходимости для приложения (Страница 1089). STEP 7 также генерирует ряд фрагментных блоков данных для сохранения представления всех HTML-страниц. При генерации блоков данных, STEP 7 обновляет свойства, чтобы отобразить номер управляющего блока данных и номер первого фрагментного блока данных. После генерации блоков данных, пользовательские веб-страницы становятся частью программы STEP 7. Блоки, соответствующие этим страницам, появляются в папке веб-сервера, которая находится в папке "Системные блоки" в программных блоках в дереве проекта.

Удаление программных блоков

Чтобы удалить сгенерированные ранее блоки данных, нажать кнопку "Удалить блоки данных". STEP 7 удаляет управляющий блок и все фрагментные блоки данных из проекта, в котором находятся пользовательские веб-страницы.

12.8.4 Конфигурирование страницы входа

В конфигурации устройства CPU можно назначить пользовательскую веб-страницу в качестве страницей входа для доступа к веб-серверу с PC или мобильного устройства. В ином случае страницей входа становится стандартнаявводная (Страница 1041) веб-страница.

Для установки пользовательской веб-страницы в качестве страницы входа, выполнить следующие действия:

1. Выбрать CPU в конфигурации устройства.
2. В окне инспектора выбрать "Веб-сервер" в свойствах CPU и активировать веб-сервер (Страница 1026).
3. В свойствах веб-сервера выбрать "Страница входа".
4. Выбрать в выпадающем списке элемент "AWP1" для конфигурирования веб-сервера таким образом, чтобы при обращении отображалась определенная пользователем страница. (При другом варианте "Вводная страница", веб-сервер при вызове отображает стандартную вводную веб-страницу.)

Кроме этого, для пользователя "Все" должно быть сконфигурировано право (Страница 1028) "Открытие пользовательских веб-страниц", а также необходимо включить вызов инструкции WWW (Страница 1089) в программу.

После завершения конфигурирования и загрузки проекта в CPU, веб-сервер может использовать "HTML-страницу по умолчанию", выбранную в качестве страницы входа при конфигурировании пользовательских веб-страниц (Страница 1087).

Примечание

Для отображения пользовательской страницы входа CPU должен находиться в рабочем состоянии RUN.

12.8.5 Программирование WWW инструкции для пользовательских веб-страниц

WWW инструкция должна быть частью программы пользователя STEP 7 и выполняться, чтобы пользовательские веб-страницы могли вызываться через стандартные веб-страницы. Управляющий блок является входным параметром WWW инструкции и определяет содержание страниц, как представлено во фрагментных блоках данных, а также информации состояния и управления. STEP 7 создает управляющий блок после нажатия на кнопку "Создать блоки" в конфигурации пользовательских веб-страниц (Страница 1087).

Программирование WWW инструкции

Программа STEP 7 должна выполнить инструкцию WWW, чтобы пользовательские веб-страницы могли вызываться через стандартные веб-страницы. Можно разрешить доступ к пользовательским веб-страницам только при определенных условиях согласно требованиям и предпочтениям приложения. В этом случае программная логика может управлять вызовом инструкции WWW.

Таблица 12-3 Инструкция WWW

LAD/FBD	SCL	Описание
	`ret_val := WWW(` `  ctrl_db:=_uint_in_);`	Доступ к пользовательским веб-страницам через стандартные веб-страницы

Необходимо ввести входной параметр управляющего блока (CTRL_DB), соответствующий целочисленному номеру DB управляющего DB. Номер управляющего DB (называемого номером веб-DB) находится в свойствах веб-сервера CPU после того, как были созданы блоки для пользовательских веб-страниц. Указать целочисленный номер DB как параметр CTRL_DB инструкции WWW. Возвращаемое значение (RET_VAL) содержит результат функции. Обратить внимание на то, что инструкция WWW выполняется асинхронно и что выход RET_VAL имеет начальное значение 0, даже если позже может произойти ошибка. Программа может проверять состояние управляющего DB, чтобы гарантировать, что приложение запустилось успешно, или через последующий вызов WWW проверить параметр RET_VAL.

Таблица 12-4 Возвращаемое значение

RET_VAL	Описание
0	Ошибки отсутствуют
16#00yx	x: Запрос, представленный соответствующим битом, находится в состоянии ожидания: x=1: Запрос 0 x=2: Запрос 1 x=4: Запрос 2 x=8: Запрос 3 Значения x могут быть логически связаны через ИЛИ, чтобы представить состояния ожидания нескольких запросов. Если, например, x = 6, то запросы 1 и 2 находятся в состоянии ожидания. y: 0: Ошибка отсутствует; 1: Ошибка имеется и "last_error" установлен в управляющем DB (Страница 1108)
16#803a	Управляющий DB не загружен.
16#8081	Управляющий DB имеет неправильный тип, формат или версию.
16#80C1	Нет доступных ресурсов для инициализации веб-приложения.

Использование управляющего DB

STEP 7 создает управляющий блок после нажатия на кнопку "Создать блоки". Номер управляющего DB отображается в свойствах пользовательских веб-страниц. Найти управляющий DB можно и в папке "Программные блоки" в дереве проекта.

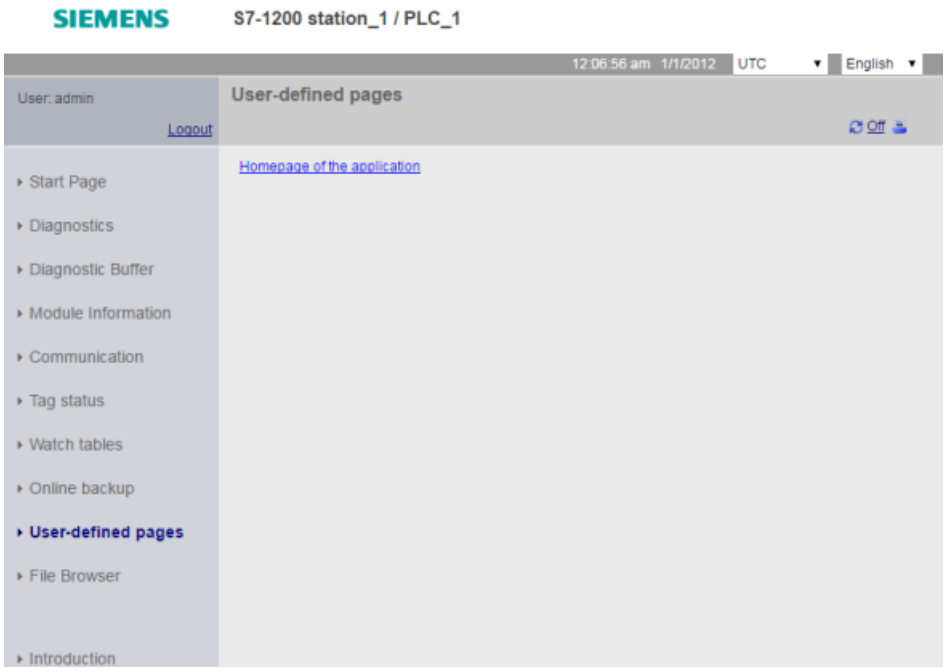
Как правило, программа STEP 7 использует управляющий DB напрямую после создания посредством "Создать блоки" без дополнительной обработки. Однако пользовательская программа STEP 7 может установить глобальные команды в управляющем DB для выключения веб-сервера или его последующего повторного включения. Кроме этого, для пользовательских страниц, которые создаются как ручные фрагментные DB (Страница 1087), пользовательская STEP 7 программа должна управлять их поведением посредством таблицы запросов в управляющем DB. Для получения информации об этих дополнительных задачах смотри тему Расширенное управление пользовательскими веб-страницами (Страница 1108).

12.8.6 Загрузка программных блоков в CPU

После генерации блоков данных для пользовательских веб-страниц, они становятся частью программы STEP 7, как и все другие программные блоки. Для загрузки программных блоков в CPU действовать как обычно. Следует помнить, что можно загрузить программные блоки для пользовательских веб-страниц, только когда CPU находится в режиме STOP.

12.8.7 Доступ к пользовательским веб-страницам

Доступ к пользовательским веб-страницам осуществляется через стандартные веб-страницы (Страница 1029). Стандартные веб-страницы отображают в навигационном меню слева ссылку "Пользовательские веб-страницы". Инструменты навигации на базовых страницах также предлагают ссылку "Пользовательские страницы". Кликом по ссылке "Пользовательские страницы" веб-браузер переходит к странице, предлагающей ссылку на страницу по умолчанию. Внутри пользовательских страниц навигация осуществляется в соответствии с тем, как были структурированы конкретные страницы.



Примечание

Также можно определить одну из пользовательских веб-страниц как страницу входа (Страница 1089) для веб-сервера.

12.8.8 Ограничения для пользовательских веб-страниц

Ограничения для стандартных веб-страниц (Страница 1112) распространяются и на пользовательские веб-страницы Кроме этого, пользовательские веб-страницы обладают некоторыми специфическими ограничениями.

Объем загружаемой памяти

После клика по "Создать блоки" пользовательские веб-страницы становятся блоками данных, которым необходимо пространство в загружаемой памяти. Если вставлена карта памяти, то доступен объем вплоть до емкости установленной карты памяти, являющейся внешней загружаемой памятью для пользовательских веб-страниц.

Если карта памяти отсутствует, эти блоки занимают внутреннюю загружаемую память, которая ограничена в соответствии с моделью CPU.

Можно проверить объем используемой и доступной загружаемой памяти с помощью интерактивных и диагностических функций в STEP 7. Также потребность в загружаемой памяти можно проверить в свойствах отдельных блоков, которые STEP 7 генерирует из пользовательских веб-страниц.

Примечание

Для уменьшения объема памяти, занимаемой пользовательскими веб-страницами, следует по возможности удалить некоторые из используемых на страницах изображений.

Кавычки в текстовых строках

Избегайте использования текстовых строк, содержащих встроенные одиночные или двойные кавычки в переменных (тегах) блока данных, на пользовательских веб-страницах. Поскольку синтаксис HTML часто использует одиночные или двойные кавычки в качестве разделителей, кавычки в текстовых строках могут помешать отображению пользовательских веб-страниц.

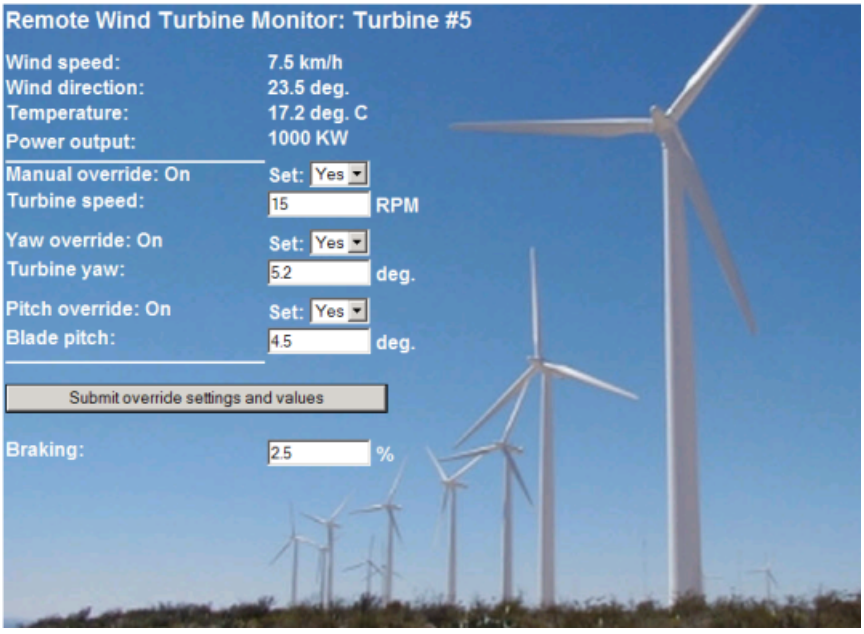
Для переменных (тегов) блока данных с типом String, которые используются на пользовательских веб-страницах, необходимо придерживаться следующих правил:

- Не использовать одиночные или двойные кавычки в строковом значении переменных (тегов) блоков данных в STEP 7.
- Не позволять пользовательской программе выполнять присвоения строк, содержащих кавычки, этим переменным (тегам) блоков данных.

12.8.9 Пример для пользовательской веб-страницы

12.8.9.1 Веб-страница для мониторинга и управления ветряной турбиной

В качестве примера пользовательской веб-страницы рассмотрим веб-страницу, которая используется, чтобы удаленно контролировать и управлять ветряной турбиной:



Описание

В этом приложении для управления каждой ветряной турбиной в ветропарке используются S7-1200. В программе STEP 7 у каждой ветряной турбины есть определенный блок данных.

Пользовательская веб-страница обеспечивает удаленный доступ к турбине через PC. Пользователь может подключиться к стандартным веб-страницам CPU определенной ветряной турбины и получить доступ к пользовательской странице "Remote Wind Turbine Monitor" (веб-страница для удаленного мониторинга ветряной турбины), чтобы просмотреть данные этой турбины. Пользователь с правами на изменение переменных может также перевести турбину в ручной режим и управлять переменными для скорости турбины, угла поворота и наклона из веб-страницы. Пользователь с правом на изменение переменных может также установить величину тормозного усилия независимо от того, находится ли турбина под ручным или автоматическим управлением.

Программа STEP 7 проверяет булевы значения для коррекции автоматического управления, и использует, если они установлены, введенные пользователями значения для скорости, ориентации и угла установки турбины. В ином случае, программа игнорирует эти значения.

Используемые файлы

Этот пример для пользовательской веб-страницы состоит из трех файлов:

- **Wind_turbine.html**: Это HTML страница, отображенная выше. С помощью AWP команд выполняется обращение к данным контроллера.
- **Wind_turbine.css**: Это - каскадная таблица стилей, которая содержит стили форматирования для HTML-страницы. Использование каскадной таблицы стилей является необязательным, но оно может упростить разработку HTML-страниц.
- **Wind_turbine.jpg**: Это - фоновое изображение, которое использует HTML-страница. Использование изображений в пользовательских веб-страницах, конечно, необязательное, и требует дополнительного объема памяти в CPU.

Этим файлы не включены в установку, но описаны в качестве примера.

Реализация

HTML-страница использует AWP-команды, чтобы выгрузить значения из PLC-устройства (Страница 1074) для полей вывода и записать в PLC значения данных (Страница 1075) из пользовательских полей ввода. Эта страница также использует AWP-команды для определения типов перечисления (Страница 1081) и для ссылки (Страница 1082) на обработку параметров настройки ВКЛ/ВЫКЛ.

Первая часть страницы представляет строку заголовка с номером ветряной турбины.



Следующая часть страницы отображает атмосферные условия на ветряной турбине. Входы/выходы на стороне турбины предоставляют информацию о скорости ветра, направлении ветра и фактической температуре.

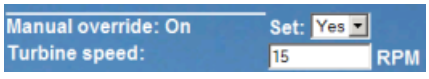


Также страница отображает выходную мощность турбины, которая считывается из S7-1200.



В следующих разделах рассматривается ручное управление турбиной, которое заменяет штатное автоматическое управление S7-1200. Возможны следующие изменения режима управления:

- Ручное управление: Включает ручное управление турбиной. Пользовательская программа STEP 7 требует установки true для ручного управления, прежде чем разрешить использование любой из ручных настроек для скорости, ориентации и угла установки турбины.



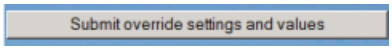
- Управление ориентацией: Включает ручное управление для ориентации турбины и ручную настройку для точной установки. Пользовательская программа STEP 7 требует установки true для ручного управления и для управления ориентацией, чтобы можно было применить параметры для ориентации.



- Регулировка угла установки: Включает ручную регулировку для угла установки лопастей. Пользовательская программа STEP 7 требует установки true для ручного управления и для регулировки угла установки, чтобы можно было применить параметры регулировки для угла установки лопастей.



HTML-страница содержит кнопку "Отправить" для передачи параметров управления контроллеру.



Поле ввода для торможения позволяет установить процентное значение для торможения вручную. Программа пользователя STEP 7 не требует ручного управления для применения значения торможения.



Кроме этого, HTML-страница использует команду AWP для записи (Страница 1079) специальной переменной, содержащей ID пользователя, который обращается к странице, в переменную (тег) в таблице переменных (тегов) PLC.

12.8.9.2 Чтение и отображение данных контроллера

HTML страница "Remote Wind Turbine Monitor" использует различные AWP команды для чтения данных из контроллера (Страница 1074) и отображения этих данных на странице. Например, можно проанализировать HTML-код для отображения выходной мощности, как показано в этой части веб-страницы:



Пример для HTML-кода

Следующий фрагмент HTML-страницы "Remote Wind Turbine Monitor" отображает текст "PowerOutput:" в левой ячейке строки таблицы, читает переменную для выходной мощности и отображает значение в правой ячейке строки таблицы вместе с текстовым сокращением для киловатт, kW.

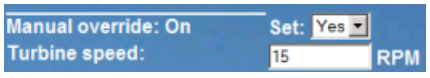
AWP-команда :="Data_block_1".PowerOutput: выполняет операцию чтения Следует обратить внимание, что ссылка на блоки данных осуществляется по имени, а не по номеру блока данных (т.е. через "Data_block_1", а не через "DB1").

```
<tr style="height:2%;">
<td>
<p>PowerOutput:</p>
```

```
</td>
<td>
<p style="margin-bottom:5px;"> :="Data_block_1".PowerOutput: kW</p>
</td>
</tr>
```

12.8.9.3 Использование типа перечисления

HTML-страница "Remote Wind Turbine Monitor" использует типы перечисления для трех мест, где на HTML-странице отображается "ON" или "OFF" для булевого значения, и там, где пользователь устанавливает булево значение. Тип перечисления для "ON" дает значение 1, а тип перечисления для "OFF" дает значение 0. Пример: Рассмотрим HTML-код для чтения и записи параметров для активации ручного управления в значении "Data_block_1".ManualOverrideEnable , используя тип перечисления:



Пример для HTML-кода

Следующие фрагменты HTML-страницы "Remote Wind Turbine Monitor" показывают, как объявить тип перечисления с именем "OverrideStatus" со значениями "Off" и "On" в 0 и 1, а затем организовать ссылку на тип перечисления "OverrideStatus" для булевой переменной (тега) "ManualOverrideEnable" в блоке данных с именем "Data_block_1".

```
<!-- AWP_In_Variable Name='"Data_block_1".ManualOverrideEnable'
Enum="OverrideStatus" -->

<!-- AWP_Enum_Def Name="OverrideStatus" Values='0:"Off",1:"On"' -->
```

Если HTML-страница содержит поле отображения в ячейке таблицы для текущего состояния "ManualOverrideEnable", она использует обычную команду чтения для переменных, но благодаря ранее объявленному и адресованному типу перечисления, на странице отображается "Off" или "On", а не 0 или 1.

```
<td style="width:24%; border-top-style: Solid; border-top-width:
2px; border-top-color: #ffffff;">
<p>Manual override: :="Data_block_1".ManualOverrideEnable:</p>
</td>
```

HTML-страница содержит раскрывающийся список для редактирования значение ManualOverrideEnable пользователем. Список выбор использует для отображения тексты "Yes" и "No". С помощью типа перечисления, текст "Yes" связан со значением "On", а текст "No" - со значением "Off". Отсутствие выбора оставляет значение ManualOverrideEnable таким как оно есть.

```
<select name='"Data_block_1".ManualOverrideEnable'>
<option value=':"Data_block_1".ManualOverrideEnable:'> </option>
<option value="On">Yes</option>
<option selected value="Off">No</option>
</select>
```

Список выбора является частью формы на HTML-странице. При нажатии пользователем на кнопку "Отправить" страница выгружает форму. При этом значение "1" записывается

в булеву переменную "ManualOverrideEnable" в Data_block_1, если пользователь выбрал "Yes", или записывается "0", если пользователь выбрал "No".

12.8.9.4 Запись введенных пользователем данных в контроллер

HTML-страница "Remote Wind Turbine Monitor" содержит несколько AWP-команд для записи данных в контроллер (Страница 1075). HTML-страница объявляет AWP_In_Variables для булевых переменных так, чтобы пользователь с правами на изменение переменных мог перевести ветряную турбину на ручное управление и разрешить ручную регулировку скорости турбины, ориентации и/или угла установки лопастей. Страница также использует AWP_In_Variables, чтобы позволить пользователю с правами на изменение переменных выполнить последующую установку значений с плавающей запятой для скорости турбины, ориентации, угла установки и процента для торможения. Страница использует HTTP команду для установки формы POST для записи AWP_In_Variables в контроллер.

Например, рассмотрим HTML-код для задания вручную значения торможения:



Пример для HTML-кода

Следующий фрагмент HTML-страницы "Remote Wind Turbine Monitor" вначале объявляет AWP_In_Variable для "Data_block_1", которая позволяет HTML-странице выполнять запись в любые переменные (теги) в блоке данных "Data_block_1". Страница отображает текст "Braking:" в левой ячейке строки таблицы. В правой ячейке строки таблицы находится поле, которое принимает введенные пользователем данные для переменной (тега) "Braking" в "Data_block_1". Это введенное значение находится в HTML-форме, которая использует метод HTTP "POST" для записи введенных текстовых данных в CPU. Затем страница считывает фактическое значение замедления из контроллера и отображает его в поле ввода данных.

Пользователь с правом на редактирование переменных может впоследствии использовать эту страницу, чтобы записать значение торможения в блок данных в CPU, который управляет торможением.

```
<!-- AWP_In_Variable Name='"Data_block_1"' -->
…
<tr style="vertical-align: top; height: 2%;">
<td style="width: 22%;"><p>Braking:</p></td>
<td>
<form method="POST">
<p><input name='"Data_block_1".Braking' size="10" type="text"> %</p>
</form>
</td>
</tr>
```

Примечание

Необходимо учитывать следующее: Если на пользовательской странице есть поле ввода данных для перезаписываемой переменной (тега) блока данных типа "String", то пользователь должен заключить строку в одиночные кавычки при вводе текста в поле.

Примечание

Обратить внимание на то, что если целый блок данных включается в объявление AWP_In_Variable, напр., <!-- AWP_In_Variable Name='"Data_block_1"' -->, тогда каждая переменная (тег) в этом блоке данных может быть записана из пользовательской веб-страницы. Этот метод следует использовать тогда, когда необходимо сделать все переменные (теги) в блоке данных перезаписываемыми. В ином случае, если необходимо, чтобы только определенные переменные (теги) блока данных были перезаписываемыми из пользовательской веб-страницы, объявить их частным образом, например, <!-- AWP_In_Variable Name='"Data_block_1".Braking' -->

12.8.9.5 Запись специальных переменных

Веб-страница "Remote Wind Turbine Monitor" записывает специальную переменную SERVER:current_user_id в переменную (тег) PLC в CPU при условии, что пользователь обладает правом на редактирование. В этом случае значение переменной (тега) PLC содержит идентификатор пользователя, который получает доступ к веб-странице "Remote Wind Turbine Monitor".

Веб-страница записывает специальную переменную в PLC без использования пользовательского интерфейса.

Пример для HTML-кода

```
<!-- AWP_In_Variable Name="SERVER:current_user_id" Use="User_ID"-->
```

12.8.9.6 Справочная информация: HTML-код веб-страницы "Remote Wind Turbine Monitor"

Wind_turbine.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<!--
Эта тестовая программа моделирует веб-страницу для мониторинга и управления
ветряной турбиной.
Необходимые переменные (теги) PLC и переменные (теги) блоков данных в STEP
7:

PLC Tag:
User_ID: Int
Data Blocks:
Data_block_1
Tags in Data_Block_1:

TurbineNumber: Int
WindSpeed: Real
WindDirection: Real
Temperature: Real
PowerOutput: Real
```

```
ManualOverrideEnable: Bool
TurbineSpeed: Real
YawOverride: Bool
Yaw: Real
PitchOverride: Bool
Pitch: Real
Braking:Real
```

Пользовательская веб-страница содержит текущие значения для данных PLC и предлагает список выбора для определения трех булевых значений с назначенным типом перечисления. С помощью кнопки "Отправить" выгружаются как выбранные булевы значения, так и поля ввода данных для скорости, ориентации и угла установки турбины. Значение торможения может быть установлено без кнопки "Отправить".

Для использования этой страницы реальная программа STEP 7 не нужна. В теории программа STEP 7 реагировала бы только на значения для скорости, ориентации и угла установки турбины, если бы были определены назначенные булевы значения. Единственной задачей STEP 7 является вызов инструкции WWW с DB-номером созданных блоков данных для этой страницы.

```
-->
<!-- AWP_In_Variable Name='"Data_block_1"' -->
<!-- AWP_In_Variable Name='"Data_block_1".ManualOverrideEnable'
Enum="OverrideStatus" -->
<!-- AWP_In_Variable Name='"Data_block_1".PitchOverride'
Enum="OverrideStatus" -->
<!-- AWP_In_Variable Name='"Data_block_1".YawOverride'
Enum="OverrideStatus" -->
<!-- AWP_In_Variable Name="SERVER:current_user_id" Use="User_ID"-->
<!-- AWP_Enum_Def Name="OverrideStatus" Values='0:"Off",1:"On"' -->

<html>
<head>
<meta http-equiv="content-type" content="text/html;
charset=utf-8"><link rel="stylesheet" href="Wind_turbine.css">
<title>Remote Wind Turbine Monitor</title>
</head>
<body>
<table cellpadding="0" cellspacing="2">
<tr style="height: 2%;">
<td colspan="2">
<h2>Remote Wind Turbine Monitor: Turbine
#:="Data_block_1".TurbineNumber:</h2>
</td>

<tr style="height: 2%;"><td style="width:
25%;"><p>WindSpeed:</p></td>
<td><p> :="Data_block_1".WindSpeed: km/h</p></td>
</tr>

<tr style="height: 2%;">
<td style="width: 25%;"><p>WindDirection:</p></td>
<td><p> :="Data_block_1".WindDirection: deg.</p></td>
```

```
</tr>

<tr style="height: 2%;"><td style="width: 25%;"><p>Temperature:</p></
td>
<td><p> :="Data_block_1".Temperature: deg. C</p></td>
</tr>

<tr style="height: 2%;">
<td style="width: 25%;"><p>PowerOutput:</p></td>
<td><p style="margin-bottom:5px;"> :="Data_block_1".PowerOutput:
kW</p>
</td>
</tr>

<form method="POST" action="">
<tr style="height: 2%;" >
<td style="width=25%; border-top-style: Solid; border-top-width:
2px; border-top-color: #ffffff;">
<p>Manual override: :="Data_block_1".ManualOverrideEnable:</p>
</td>
<td class="Text">Set:

<select name='"Data_block_1".ManualOverrideEnable'>
<option value=':="Data_block_1".ManualOverrideEnable:'> </option>
<option value="On">Yes</option>
<option value="Off">No</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;"><td style="width:
25%;"><p>TurbineSpeed:</p></td>
<td>
<p style="margin-bottom:5px;"><input
name='"Data_block_1".TurbineSpeed' size="10"
value=':="Data_block_1".TurbineSpeed:' type="text"> RPM</p>
</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;">
<p>YawOverride: :="Data_block_1".YawOverride: </p>
</td>
<td class="Text">Set:

<select name='"Data_block_1".YawOverride'>
<option value=':="Data_block_1".YawOverride:'> </option>
<option value="On">Yes</option>
<option value="Off">No</option>
</select>
```

```
</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;">
<p>Turbine yaw:</p>
</td>
<td>
<p style="margin-bottom:5px;"><input name='"Data_block_1".Yaw'
size="10" value=':="Data_block_1".Yaw:' type="text"> deg.</p>
</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;">
<p>PitchOverride: :="Data_block_1".PitchOverride: </p>
</td>
<td class="Text">Set:

<select name='"Data_block_1".PitchOverride'>
<option value=':="Data_block_1".PitchOverride:'> </option>
<option value="On">Yes</option>
<option value="Off">No</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width=25%; border-bottom-style: Solid; border-bottom-
width: 2px; border-bottom-color: #ffffff;">
<p>Blade pitch:</p>
</td>
<td>
<p style="margin-bottom:5px;"><input name='"Data_block_1".Pitch'
size="10" value=':="Data_block_1".Pitch:' type="text"> deg.</p>
</td>

</tr>
<tr style="height: 2%;">
<td colspan="2">
<input type="submit" value="Отправить параметры и значения для
управления">
</td>
</tr>
</form>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;"><p>Braking:</p></td>
<td>
<form method="POST" action="">
```

```
<p> <input name='"Data_block_1".Braking' size="10"
value=':="Data_block_1".Braking:' type="text"> %</p>
</form>
</td>
</tr>
<tr><td></td></tr>

</table>
</body>
</html>
```

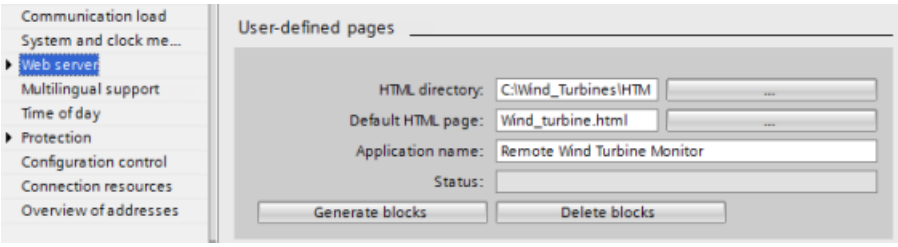
Wind_turbine.css

```
BODY {
    background-image: url('./Wind_turbine.jpg');
    background-position: 0% 0%;
    background-repeat: no-repeat;
    background-size: cover;
}
H2 {
    font-family: Arial;
    font-weight: bold;
    font-size: 14.0pt;
    color: #FFFFFF;
    margin-top:0px;
    margin-bottom:10px;
}
P {
    font-family: Arial;
    font-weight: bold;
    color: #FFFFFF;
    font-size: 12.0pt;
    margin-top:0px;
    margin-bottom:0px;
}
TD.Text {
    font-family: Arial;
    font-weight: bold;
    color: #FFFFFF;
    font-size: 12.0pt;
    margin-top:0px;
    margin-bottom:0px;
}
```

12.8.9.7 Пример конфигурирования веб-страницы в STEP 7

Чтобы вставить HTML-страницу "Remote Wind Turbine Monitor" как пользовательскую веб-страницу для S7-1200, необходимо сконфигурировать данные для HTML-страницы в STEP 7 и создать блоки данных из HTML-страницы.

Обратиться к свойствам CPU для S7-1200, который управляет ветряной турбиной, и ввести конфигурационную информацию в свойствах пользовательских страниц веб-сервера:



Поля конфигурирования

- HTML каталог: Это поле определяет полный путь к папке на компьютере, где расположена страница по умолчанию (домашняя страница или начальная страница). Кнопка "..." позволяет найти путь к папке, которая необходима.
- HTML страница по умолчанию: Это поле указывает имя файла страницы по умолчанию или домашней страницы HTML-приложения. Кнопка "..." позволяет выбрать необходимый файл. Для этого примера WindTurbine.html является HTML-страницей по умолчанию. Пример по удаленному контролю ветряной турбины "Remote Wind Turbine Monitor" состоит только из одной страницы, но в других пользовательских приложениях страница по умолчанию может вызывать дополнительные страницы с помощью ссылок. В HTML-коде страница по умолчанию должна ссылаться на другие страницы относительно исходной HTML-папки.
- Имя приложения: Это дополнительное поле содержит имя, которое веб-браузер отображает в поле адреса при вызове страницы. Для данного примера - это "Remote Wind Turbine Monitor", но можно использовать любое имя.

Никакие другие поля не требуют конфигурации.

Заключительные шаги

Чтобы пользоваться HTML-страницей "Remote Wind Turbine Monitor" согласно её конфигурации, следует создать блоки, запрограммировать инструкцию WWW (Страница 1089) с номером сгенерированного управляющего DB в качестве входного параметра, загрузить программные блоки в CPU и перевести CPU в режим RUN.

Когда оператор впоследствии обращается к стандартным веб-страницам для S7-1200, который управляет ветряной турбиной, веб-страница "Remote Wind Turbine Monitor" доступна по ссылке "Пользовательские страницы" на панели навигации. Теперь эта страница содержит инструменты для мониторинга и управления ветряной турбиной.

12.8.10 Создание пользовательских веб-страниц на нескольких языках

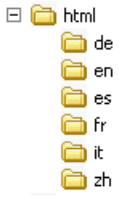
Веб-сервер предоставляет средства для создания пользовательских веб-страниц на следующих языках:

- Немецкий (de)
- Английский (en)
- Испанский (es)
- Французский (fr)
- Итальянский (it)
- Упрощенный китайский (zh)

С этой целью HTML-страницы создаются в структуре папок (Страница 1105), которая соответствует языкам, и устанавливается специальный куки "siemens_automation_language" для страниц пользователя (Страница 1105). Веб-сервер реагирует на этот куки и переключается на страницу по умолчанию в соответствующей языковой папке.

12.8.10.1 Создание структуры папок

Для доступности пользовательских веб-страниц на нескольких языках, в HTML-каталоге создается структура папок. Имена папок должны состоять из двух букв и выбираются согласно правилам, показанным ниже:



de: немецкий
en: английский
es: испанский
fr: французский
it: итальянский
zh: упрощенный китайский

На том же уровне могут размещаться и любые другие папки для страниц, например, папки для изображений или скриптов.

Можно вставлять любые языковые папки. Не обязательно использовать все шесть языков. В языковых папках HTML-страницы создаются и программируются на соответствующем языке.

12.8.10.2 Программирование переключения языка

Веб-сервер выполняет переключение между языками с помощью куки с именем "siemens_automation_language". Этот куки, определенный и установленный на HTML-страницах, интерпретируется веб-сервером, чтобы отобразить страницу на соответствующем языке из языковой папки с тем же именем. HTML-страница должна содержать JavaScript, чтобы установить этот куки на один из заранее определенных идентификаторов языка: "de", "en", "es", "fr", "it', или "zh".

Например, если HTML-страница устанавливает куки на "de", то веб-сервер переключается на папку "de" и отображает страницу с именем HTML-страницы по умолчанию, как определено в конфигурации STEP 7 (Страница 1108).

Пример

Следующий пример использует HTML-страницу по умолчанию с именем "langswitch.html" в каждой из языковых папок. Также в HTML-каталоге находится папка с именем "script". Папка "script " содержит файл JavaScript-Datei с именем "lang.js". Каждая страница "langswitch.html " использует этот JavaScript для установки языкового куки "siemens_automation_language".

HTML для "langswitch.html" в папке "en"

Заголовок HTML-страницы устанавливает язык на английский, выбирает набор символов UTF-8 и задает путь к JavaScript файлу "lang.js".

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="en">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Language switching english page</title>
<script type="text/javascript" src="script/lang.js" ></script>
```

В содержимом файла есть список выбор для переключения между немецким и английским языком. Английский ("en") является предварительно выбранным языком. Если пользователь изменяет язык, страница вызывает JavaScript функцию DoLocalLanguageChange () со значением выбранной опции.

```
<!-- Language Selection -->
<table>
   <tr>
      <td align="right" valign="top" nowrap>
       <!-- change language immediately on selection change -->
          <select name="Language"
                    onchange="DoLocalLanguageChange(this)"
                    size="1">
            <option value="de" >German</option>
            <option value="en" selected >English</option>
           </select>
      </td>
   </tr>
</table><!-- Language Selection End-->
```

HTML для "langswitch.html" в папке "de"

Заголовок для немецкой страницы langswitch.html такой же, как и для английского языка, за исключением установки на немецкий язык.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="de"><meta http-
equiv="Content-Type" content="text/html; charset=utf-8">
<title>Sprachumschaltung Deutsche Seite</title>
<script type="text/javascript" src="script/lang.js" ></script>
</head>
```

HTML код на немецкой странице идентичен коду английской страницы, за исключением того, что языком по умолчанию является немецкий ("de").

```
<!-- Language Selection -->
<table>
   <tr>
     <td align="right" valign="top" nowrap>
      <!-- change language immediately on change of the selection -->
       <select name="Language"
               onchange="DoLocalLanguageChange(this)"
               <size="1">
            <option value="de" selected >Deutsch</option>
            <option value="en" >Englisch</option>
       </select>
      </td>
   </tr>
</table><!-- Language Selection End-->
```

JavaScript "lang.js" в папке "script"

Функция "DoLocalLanguageChange()" находится в файле "lang.js". Эта функция вызывает функцию "SetLangCookie()" и после обновляет окно, в котором отображается HTML-страница.

Функция "SetLangCookie()" присваивает куки siemens_automation_language" документа значение из списка выбора. Кроме этого, функция определяет путь приложения, поэтому значение куки получает переключенная страница, а не запрашивающая страница.

Дополнительно, в разделе комментариев, страница может указать значение для истечения срока действия куки.

```
function DoLocalLanguageChange(oSelect) {
        SetLangCookie(oSelect.value);
        top.window.location.reload();
    }
function SetLangCookie(value) {
        var strval = "siemens_automation_language=";
        // Это куки, с помощью которого веб-сервер
    // определяет требуемый язык.
    // Это имя необходимо для веб-сервера.
        strval = strval + value;
        strval = strval + "; path=/ ;";
        // Указать путь для приложения, иначе
    // путь был бы передан на вызывающую страницу
    // и эта страница не получила бы куки.
        /* OPTIONAL
           Указать срок действия, если куки должен существовать дольше
      чем текущий сеанс браузера:
           var now      = new Date();
           var endttime = new Date(now.getTime() + expiration);
           strval = strval + "; expires=" +
                    endttime.toGMTString() + ";";
        */
```

```
        document.cookie = strval;
    }
```

Примечание

Если пользовательская веб-страница содержит HTML-файлы в определенных языковых папках (например en, de), а также HTML-файлы, которые не находятся в определенных языковых папках, то с помощью команды AWP_Enum_Def нельзя определить типы перечисления в файлах, находящихся в обоих местах. Вместо этого, при использовании типов перечисления, они должны быть определены в файлах в или вне определенных языковых папок. Нельзя выполнить объявление перечислений в файлах обоих мест хранения.

12.8.10.3 Конфигурирование STEP 7 для использования многоязыковой структуры страницы

Процесс конфигурирования многоязыковых пользовательских веб-страниц аналогичен процессу конфигурирования пользовательских веб-страниц (Страница 1087). Если же были созданы папки для языков, то следует указать в параметрах HTML-каталога папку, которая содержит отдельные языковые папки. HTML-каталог не устанавливается в качестве одной из языковых папок.

При выборе HTML-страницы по умолчанию, выполняется переход в языковую папку для выбора HTML-страницы, которая должна быть начальной страницей. Впоследствии при генерации блоков и загрузке блоков в CPU, веб-сервер отображает начальную страницу из сконфигурированной языковой папки.

Пример: Если структура папок, показанная здесь, находится на C:\, настройкой для HTML-каталога является C:\html. А если сначала должна отображаться страница на английском языке, то перейти для настройки HTML-страницы по умолчанию по следующему пути en\langswitch.html.



12.8.11 Расширенное управление пользовательскими веб-страницами

При генерации блоков данных для пользовательских веб-страниц, STEP 7 создает управляющий DB, который он использует для управления отображением и взаимодействием с пользовательскими страницами. STEP 7 также создает набор фрагментных DB, которые представляют отдельные страницы. При нормальных условиях знания структуры управляющего DB не требуется, как и умения его редактировать.

Если, например, необходимо включать или выключать веб-приложение или изменять отдельные фрагменты вручную, то для этого можно использовать переменные (теги) управляющего DB и инструкцию WWW.

Структура управляющего DB

Управляющий DB имеет большую структуру данных, и доступен при программировании пользовательской программы STEP 7. Здесь описаны лишь некоторые из переменных (тегов) управляющего блока.

Структура Commandstate

"Commandstate" является структурой, которая содержит глобальные команды и глобальные состояния для веб-сервера.

Глобальные команды в структуре "Commandstate"

Глобальные команды применяются к веб-серверу в целом. Можно деактивировать веб-сервер или перезапустить его с помощью параметров управляющего DB.

Переменная (тег) блока	Тип данных	Описание
init	BOOL	Анализ управляющего DB и инициализация веб-приложения
deactivate	BOOL	Деактивация веб-приложения

Глобальные состояния в структуре "Commandstate"

Глобальные состояния применяются к веб-серверу в целом и содержат информацию о статусе веб-приложения.

Переменная (тег) блока	Тип данных	Описание
initializing	BOOL	Веб-приложение читает управляющий DB
error	BOOL	Инициализация веб-приложения не может быть выполнена
deactivating	BOOL	Веб-приложение завершается
deactivated	BOOL	Веб-приложение завершено
initialized	BOOL	Веб-приложение инициализировано
last_error	INT	Последняя ошибка, возвращенная вызовом инструкции WWW (Страница 1089), когда код возврата для WWW равен 16#0010: 16#0001: Противоречивая структура фрагментного DB 16#0002: Имя приложения уже существует 16#0003: Отсутствуют ресурсы (память) 16#0004: Противоречивая структура управляющего DB 16#0005: Фрагментный DB недоступен 16#0006: Фрагментный DB не для AWP 16#0007: Противоречивые данные перечисления 16#000D: Конфликт из-за размера управляющего DB

Таблица запросов

Таблица запросов представляет собой массив структур, содержащих команды и состояния, которые применяются к отдельному фрагментному DB. После создания

фрагментов "ручного" типа с помощью команды AWP_Start_Fragment (Страница 1083), пользовательская программа STEP 7 должна управлять этими страницами с помощью управляющего DB. Состояния запроса имеют статус только для чтения и предоставляют информацию о текущем фрагменте. Команды запроса используются для управления текущим фрагментом.

Переменная (тег) блока	**Тип данных**	**Описание**
requesttab	ARRAY [1 .. 4] OF STRUCT	Массив структур для управления отдельными фрагментными DB Веб-сервер может обработать до четырех фрагментов одновременно. Индекс массива для определенного фрагмента выбирается произвольно, когда веб-сервер обрабатывает несколько фрагментов или фрагменты из нескольких сеансов браузера.

Компоненты структуры requesttab

Переменная (тег) блока	**Тип данных**	**Описание**
page_index	UINT	Номер текущей веб-страницы
fragment_index	UINT	Номер текущего фрагмента - возможно определение другого фрагмента
// Команды запроса		
continue	BOOL	Разрешает передачу текущей страницы / фрагмента с продолжением на следующем фрагменте
repeat	BOOL	Разрешает повторную передачу текущей страницы / фрагмента с продолжением на том же фрагменте
abort	BOOL	Закрыть http соединение без передачи
finish	BOOL	Передать этот фрагмент; страница завершена – не обрабатывать дополнительные фрагменты
// Состояния запроса		Состояния запроса имеют статус только для чтения.
idle	BOOL	Ничего не делать, но оставаться активным
waiting	BOOL	Фрагмент ожидает разрешения
sending	BOOL	Передача фрагмена
aborting	BOOL	Пользователь отменил текущий запрос

Принцип работы

Каждый раз, когда программа вносит изменения в управляющий DB, она должна вызывать инструкцию WWW с номером измененного управляющего DB в качестве параметра. Глобальные команды и команды запроса вступают в силу, когда пользовательская программа STEP 7 выполняет инструкцию WWW (Страница 1089).

Пользовательская программа STEP 7 может установить fragment_index явно, заставив тем самым веб-сервер обработать указанный фрагмент по команде запроса. В ином случае, веб-сервер обрабатывает актуальный фрагмент для текущей страницы, когда выполняется инструкция WWW.

Возможные методы для использования fragment_index:

- Обработка текущего фрагмента: Оставить fragment_index без изменений и установить команду продолжения.
- Пропуск текущего фрагмента: Установить fragment_index на 0 и установить команду продолжения.
- Замена текущего фрагмента на другой фрагмент: Указать для fragment_index новый ID фрагмента и установить команду продолжения

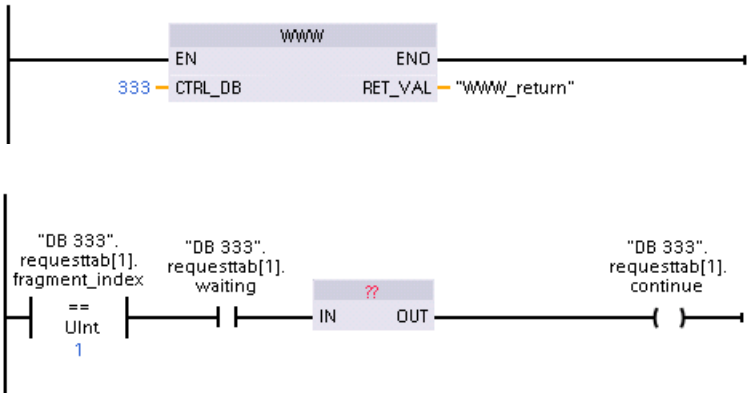
Чтобы проверить глобальные состояния или состояния запроса, которые, возможно, изменились, пользовательская программа STEP 7 должна вызвать инструкцию WWW, чтобы проанализировать текущие значения этих состояний. Типовым использованием может быть периодический вызов инструкции WWW, пока не наступит определенное состояние.

Примечание

Если пользовательская программа STEP 7 устанавливает несколько команд запроса, то инструкция WWW обрабатывает только одну из них, придерживаясь следующей очередности: отмена, завершение, повтор, продолжение. Инструкция WWW удаляет все команды запроса после обработки.

Примеры

В следующем примере показана пользовательская программа STEP 7, которая проверяет, находится ли фрагмент с идентификатором 1 в состоянии ожидания после предварительного вызова инструкции WWW. Это может быть и ожиданием других определенных прикладных условий. Затем она выполняет необходимую для фрагмента обработку, такую как установка переменных (тегов) блоков данных, выполнение вычислений или другие специализированные задачи. Затем она устанавливает флаг продолжения, чтобы веб-сервер выполнил этот фрагмент.



Когда программа вызывает инструкцию WWW с этим измененным управляющим DB, пользовательская веб-страница с этим фрагментом может быть отображена веб-браузером.



Следует учитывать, что это упрощенный пример. Проверяемый фрагмент может находиться в любой из четырех структур requesttab массива. Программа должна обработать все четыре структуры requesttab.

12.9 Ограничения

Следующие факторы, имеющие отношение к информационным технологиям, могут влиять на использование веб-сервера:

- Как правило, для доступа к стандартным веб-страницам или пользовательским веб-страницам необходимо использовать IP-адрес CPU или IP-адрес беспроводного маршрутизатора с номером порта. Если веб-браузер не разрешает прямые соединения по IP-адресу, то следует обратиться к системному администратору. Если локальная политика поддерживает DNS, то можно подключиться к IP-адресу через запись DNS по этому адресу.
- Брандмауэры, настройки прокси-сервера и другие локальные ограничения также могут препятствовать доступу к CPU. Обратиться к своему системному администратору для решения этих проблем.
- Стандартные веб-сайты используют Javascript и куки. Если настройки веб-браузера запрещают JavaScript или куки, то следует изменить их. Если изменение невозможно, то некоторые функции будут ограничены (Страница 1113). Использование JavaScript и куки на пользовательских веб-страницах не является обязательным. Если они используются, то должны быть активированы в браузере.
- Веб-сервер поддерживает протокол SSL. Можно получить доступ к стандартным веб-страницам и пользовательским веб-страницам по одному из двух URL: http:// ww .xx.yy.zz или https://ww.xx.yy.zz, где "ww.xx.yy.zz". "ww.xx.yy.zz" - это IP-адрес CPU.
- Siemens предлагает сертификат безопасности для безопасного доступа к веб-серверу. На стандартной вводной веб-странице (Страница 1041) можно загрузить сертификат и импортировать его в интернет-опции браузера (Страница 1033). Если не импортировать сертификат, то приглашение на проверку безопасности будет появляться каждый раз при обращении к веб-серверу по https://.

Количество соединений

Веб-сервер поддерживает до 30 активных соединений. Эти 30 соединений могут использоваться различным способом, в зависимости от задействованного веб-браузера и числа различных объектов на странице (.css файлы, изображения, другие .html файлы). Некоторые соединения сохраняются, пока веб-сервер отображает страницу; другие разрываются после первоначального соединения.

Например, при использовании определенных версии Mozilla Firefox, которые поддерживают максимум шесть постоянных соединений, можно использовать пять экземпляров браузера или вкладок браузера, прежде чем веб-сервер начнет разрывать соединения. Если страница не использует все шесть соединений, можно открывать дополнительные экземпляры браузера или вкладки браузера.

Также следует иметь в виду, что количество активных соединений может повлиять на скорость загрузки страницы.

Примечание

Выход из системы перед закрытием веб-сервера

Если был выполнен вход в систему на веб-сервере, то необходимо обязательно выйти из системы перед закрытием веб-браузера. Веб-сервер поддерживает максимум семь параллельных авторизаций.

Если выход из системы не будет выполнен, то, в зависимости от браузера, может остаться несколько одновременно открытых соединений. Открывая и закрывая окна браузера веб-сервера несколько раз без выхода из системы, можно использовать все 30 соединений. Если все соединения будут использованы, то при следующей попытке входа в систему появится сообщение "Неверный логин". Придется подождать до 30 минут, прежде чем веб-сервер освободит достаточное число соединений, чтобы можно было снова войти в систему. Чтобы избежать этой проблемы, следует всегда выходить из системы, прежде чем закрывать веб-сервер (если вход был выполнен).

12.9.1 Использование JavaScript

Стандартные веб-страницы используют HTML, JavaScript и куки. Если сайт ограничивает использование JavaScript и куки, то следует разрешить их для правильного функционирования страниц. Если нельзя включить JavaScript для веб-браузера, то стандартные веб-страницы не будут работать. Можно использовать базовые страницы, для которых JavaScript не требуется.

См. также

Компоновка стандартных веб-страниц (Страница 1036)

12.9.2 Ограничения функциональности, когда интернет-опции отключают куки

Если отключить куки в веб-браузере, то будут иметь место следующие ограничения:

- Вход в систему невозможен.
- Переключение языка невозможно.
- Изменение времени с UTC на PLC невозможно. Без куки действует только UTC время.

12.9.3 Правила ввода имен переменных (тегов) и значений

Обратить внимание на следующие моменты при использовании стандартной страницы Состояние переменных (тегов) (Страница 1055)иТаблицы текущего контроля (Страница 1056):

- При изменении всего значения DTL переменной (тега), напр., "Data_block_1_.DTL_tag, использовать следующий синтаксис DTL для измененного значения: DTL#JJJJ-MM-TT-HH-MM-SS[.ssssssss]
- При вводе значения с типом данных Real или LReal в экспоненциальном представлении, действовать следующим образом:
  - Для ввода вещественного значения (Real или LReal) с положительным показателем степени (например, +3.402823e+25), ввести это значение в одном из следующих форматов:
    +3.402823e25
    +3.402823e+25
  - Для ввода вещественного значения (Real или LReal) с отрицательным показателем степени (например, +3.402823e-25), ввести это значение следующим образом:
    +3.402823e-25
  - Убедиться, что часть мантиссы вещественного значения в экспоненциальной записи включает десятичную точку. Если опустить десятичную точку, то значение изменится в целочисленное значение. Пример: Следует ввести -1.0e8, а не -1e8.
- Значения LReal могут содержать только 15 цифр (независимо от расположения десятичной точки). Ввод более 15 цифр создает ошибку округления.

Ограничения на странице Состояние переменных (тегов) и Таблица текущего контроля:

- Максимальное количество символов для URL составляет 2083. URL, соответствующей текущей странице, можно увидеть в адресной строке браузера.
- Что касается формата отображения символов, если после обработки браузером фактические значения CPU не являются действительными символами ASCII, на странице отображается символ, которому предшествует знак доллара: $.

12.9.4 Импорт архивов данных в формате CSV в не американские/английские версии Microsoft Excel

Архивные файлы сохраняются в формате CSV (разделенные запятой значения). Можно открывать эти файлы со страницы архивов данных непосредственно в Excel, если в системе установлена американская или английская версия Excel. В других странах, однако, этот формат не используется широко, потому что запятые часто встречаются в системе представления чисел.

Если используется не американская или английская версия Excel, то выполнить следующие действия, чтобы открыть сохраненный файл архива данных:

1. Открыть Microsoft Excel и создать пустую рабочую папку.
2. Выбрать в меню "Данные > Импорт внешних данных" команду "Импорт данных".
3. Перейти к архивному файлу, который необходимо открыть, и выбрать его. Мастер импорта текста запускается.

4. В мастере импорта текста изменить параметр по умолчанию для "Тип данных оригинала" с "Фиксированная ширина" на "С разделителями".
5. Нажать кнопку "Дальше".
6. В диалоговом окне "Шаг 2" установить флажок "Запятая", чтобы изменить тип разделителя с табулятора на запятую.
7. Нажать кнопку "Дальше".
8. В диалоговом окне "Шаг 3" при желании можно изменить формат даты с MDY (месяц / день / год) на другой формат.
9. Выполнить оставшиеся шаги мастера импорта текста, чтобы импортировать файл.

Коммуникационный процессор и Modbus TCP 13

13.1 Использование последовательных коммуникационных интерфейсов

Два коммуникационных модуля (CM) и одна коммуникационная плата (CB) оснащены интерфейсом для PtP-коммуникаций:

- CM 1241 RS232 (Страница 1621)
- CM 1241 RS422/485 (Страница 1622)
- CB 1241 RS485 (Страница 1619)

Возможно подключение до трех CM (любого типа) плюс CB, т.е. в общей сложности четыре коммуникационных интерфейса. Установить CM слева от CPU или другого CM. CB вставляется с лицевой стороны CPU. Обратиться к инструкциям по установке (Страница 66) для получения информации о монтаже и демонтаже модулей.

Последовательные коммуникационные интерфейсы имеют следующие характеристики:

- Изолированный порт
- Поддержка протоколов "точка-точка"
- Конфигурирование и программирование через инструкции коммуникационного процессора "точка-точка"
- Отображение активности, связанной с передачей и приемом, посредством светодиодов
- Светодиодная диагностика (только CM)
- Получение питания от CPU: не требуется внешний источник питания.

См. технические данные для коммуникационных интерфейсов (Страница 1609)

Светодиодная индикация

У коммуникационных модулей есть три светодиодных индикатора:

- Диагностический светодиод (DIAG): Этот светодиод мигает красным цветом до тех пор, пока к нему не обратится CPU. После включения CPU он проверяет наличие CM и обращается к ним. Диагностический светодиод начинает мигать зеленым цветом. Это означает, что CPU обратился к CM, но еще не назначил ему конфигурацию. CPU загружает конфигурацию в настроенные CM, при загрузке программы в CPU. После загрузки в CPU диагностический светодиод на коммуникационном модуле должен светиться зеленым в непрерывном режиме.
- Светодиод передачи (Tx): Светодиод передачи светится при отправке данных через коммуникационный порт.
- Светодиод приема (Rx): Светодиод приема светится при получении данных через коммуникационный порт.

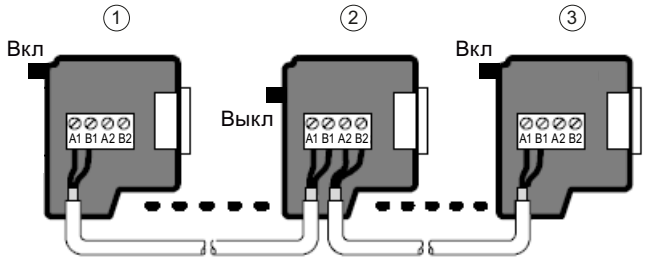
Коммуникационная плата предлагает светодиоды передачи (TxD), и приема (RxD). У нее нет диагностического светодиода.

13.2 Включение оконечной нагрузки в шинном разъеме RS485

Siemens предлагает шинный соединительный разъем RS485 (Страница 1639), который можно использовать для простого подключения нескольких устройств к сети RS485. Разъем имеет две группы клемм, которые позволяют подключать входящие и исходящие сетевые кабели. Разъем также оборудован переключателем для выборочного включения оконечной нагрузки.

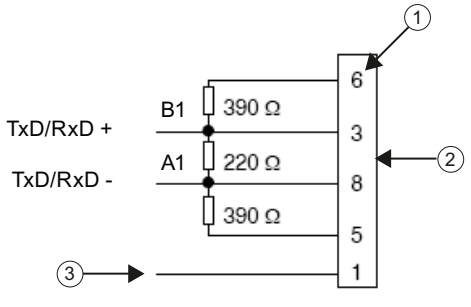
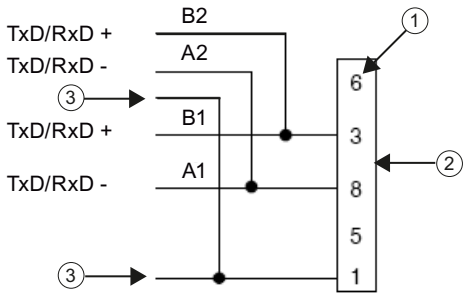
Примечание

Оконечная нагрузка включается только для двух концов сети RS485. Оконечная нагрузка не включается для устройств между обоими оконечными устройствами. Экран разделанного кабеля: должен контактировать с металлическими направляющими в местах всех подключений на участке в 12 мм.



① Положение переключателя = Вкл: оконечная нагрузка подключена
② Положение переключателя = Выкл: оконечная нагрузка не подключена
③ Положение переключателя = Вкл: оконечная нагрузка подключена

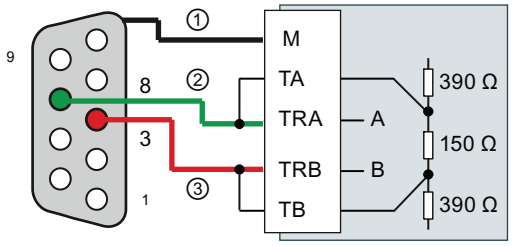
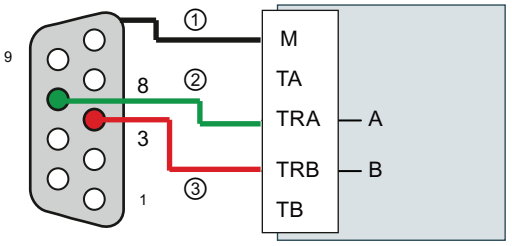
Таблица 13-1 Оконечная нагрузка для разъема RS485

Оконечное устройство (оконечная нагрузка ВКЛ)	**Не оконечное устройство (оконечная нагрузка ВЫКЛ)**
	

① Номер контакта
② Шинный разъем
③ Экран кабеля

CB 1241 предлагает внутренние резисторы в качестве терминаторов для сети. Для включения оконечной нагрузки для соединения, подключить TRA к TA и TRB к TB для включения внутренних резисторов в схему. У CB 1241 нет 9-контактного разъема. В следующей таблице показаны подключения к 9-контактному разъему на коммуникационном партнере.

Таблица 13-2 Оконечная нагрузка для CB 1241

Оконечное устройство (оконечная нагрузка ВКЛ)	**Не оконечное устройство (оконечная нагрузка ВЫКЛ)**
	

① Соединить M с экраном кабеля
① A = TxD/RxD - (зеленый провод / контакт 8)
② B = TxD/RxD + (красный провод / контакт 3)

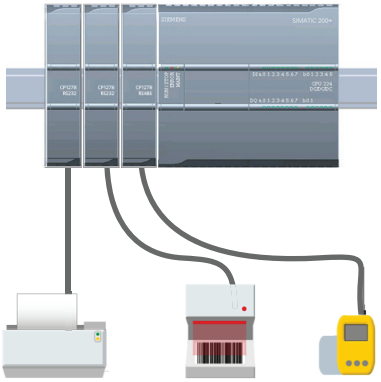
13.3 Коммуникации точка-точка (PtP)

CPU поддерживает следующие типы коммуникации "точка-точка" (PtP) для символьно-ориентированных последовательных протоколов.

- PtP, свободно программируемая коммуникация (Страница 1120)
- PtP, 3964(R) (Страница 1122)
- USS (Страница 1176)
- Modbus (Страница 1195)

13.3.1 PtP, свободно программируемая коммуникация

Протокол двухточечного соединения PtP для свободно программируемой коммуникации обеспечивает максимум свободы и гибкости, но требует значительных усилий при реализации в программе пользователя.



PtP предлагает большой выбор возможностей:

- Прямая передача данных на внешнее устройство, например, принтер
- Прием данных от других устройств, например, считывателей штрих-кода, считывателей устройств высокочастотной идентификации (RFID), видеокамер и систем технического зрения других фирм и многих других типов устройств
- Обмен информацией с другими устройствами в режиме приема/передачи, например, устройствами GPS, видеокамерами или системами технического зрения других фирм, радиомодемами и т. д.

PtP-коммуникация обеспечивает последовательный обмен данными с помощью стандартных универсальных асинхронных приемопередатчиков (UART), которые поддерживают различные скорости передачи и контроль четности. RS232 и RS422/485 коммуникационные модули (CM 1241) и коммуникационная плата RS485 (CB 1241) имеют электрические интерфейсы для выполнения PtP-коммуникаций.

Свободно программируемая PtP коммуникация по PROFIBUS или PROFINET

С помощью PtP можно использовать распределенную стойку ввода-вывода PROFINET или PROFIBUS для связи с различными устройствами (RFID-считыватели, устройство GPS и другие):

- PROFINET (Страница 810): Интерфейс Ethernet S7-1200 CPU подключается к интерфейсному модулю PROFINET. Коммуникационные модули PtP в стойке с интерфейсным модулем могут в этом случае обеспечить последовательную коммуникацию с PtP-устройствами.
- PROFIBUS (Страница 984): Коммуникационный модуль PROFIBUS устанавливается на левой стороны стойки с S7-1200 CPU. Коммуникационный модуль PROFIBUS соединяется со стойкой, содержащей интерфейсный модуль PROFIBUS. Коммуникационные модули PtP в стойке с интерфейсным модулем могут в этом случае обеспечить последовательную коммуникацию с PtP-устройствами.

Поэтому S7-1200 поддерживает два набора PtP-инструкций:

- Старые инструкции точка-точка (Страница 1297): Эти инструкции использовались до версии V4.0 S7-1200 и обеспечивают последовательную коммуникацию только с использованием коммуникационного модуля CM 1241 или коммуникационной платы CB 1241.
- Инструкции точка-точка (Страница 1139): Эти инструкции обеспечивают всю функциональность старых инструкций плюс возможность подключения PtP коммуникационных модулей через распределенный ввод-вывод PROFINET и PROFIBUS. Инструкции "точка-точка" позволяют обращаться к коммуникационным модулям через стойку распределенного ввода-вывода.
  Для использования инструкций точка-точка S7-1200 модулям CM 1241 потребуется как минимум прошивка версии V2.1. Эти модули могут подключаться только в локальной стойке с левой стороны S7-1200 CPU. Инструкции точка-точка могут использоваться и для CB 1241.
  При обмене данными через распределенную периферию используются следующие модули:

Станция	Модуль	Заказной номер	Интерфейс
ET 200MP	CM PtP RS232 BA	6ES7540-1AD00-0AA0	RS232
	CM PtP RS232 HF	6ES7541-1AD00-0AB0	RS232
	CM PtP RS422/485 BA	6ES7540-1AB00-0AA0	RS422/RS485
	CM PtP RS422/485 HF	6ES7541-1AB00-0AB0	RS422/RS485
ET 200SP	CM PtP	6ES7137-6AA00-0BA0	RS232 и RS422/RS485

Примечание

С помощью инструкции точка-точка можно обращаться к коммуникационной плате, локальным последовательным модулям (на левой стороне), к последовательным модулям через PROFINET и к последовательным модулям через PROFIBUS. STEP 7 предлагает старые PtP инструкции только для поддержания имеющихся программ. Но старые инструкции работают и на новых S7-1200 CPU. Преобразование старых программ под новые инструкции не требуется.

Примечание

Требование версии прошивки модуля CM для синхронизации времени и PtP-коммуникации

Если опция "CPU синхронизирует модули устройства" включена в свойствах Синхронизация времени (Страница 176) для интерфейса Profinet в конфигурации устройства, то необходимо обновить прошивку подключенных коммуникационных модулей до последней версии. Разрешение на синхронизацию времени для коммуникационных модулей со старыми версиями прошивки может вызвать ошибки связи.

13.3.2 3964(R)-коммуникация

S7-1200 CPU поддерживает протокол 3964(R) для обмена данными между CM 1241 модулем (RS-232) или CM 1241 модулем (RS-422/485) и партнером, использующим протокол 3964(R). В отличие от описанной выше PtP коммуникации с определенными конкретными характеристиками отправки (передачи) и приема сообщений, протокол 3964(R) использует строгий протокол со следующими управляющими символами:

- STX Start of Text
  Начало передаваемой последовательности символов
- DLE Data Link Escape
  Переключение передачи данных
- ETX End of Text
  Конец передаваемой последовательности символов
- BCC Block Check Character – символ проверки блока
- NAK Negative Acknowledge – отрицательный ответ

Полное описание протокола можно найти в главе руководства Справочник S7-300 двухточечное соединение CP 341 создание и параметрирование. (https://support.industry.siemens.com/cs/us/en/view/1117397) с описанием принципов последовательной передачи данных.

Конфигурирование коммуникационного модуля

Для обмена данными с партнером по протоколу 3964(R), необходимо включить в конфигурацию устройства в STEP 7 один из следующих коммуникационных модулей:

- CM 1241 (RS-232)
- CM 1241 (RS-422/485)

CM модуль должен иметь версию прошивки V2.2.0 или выше.

После для коммуникационного модуля необходимо сконфигурировать коммуникационные порты (Страница 1123), приоритет и параметры протокола (Страница 1137).

Коммуникация с партнером по протоколу 3964(R)

Если CM конфигурируется для протокола 3964(R), то следует использовать стандартные инструкции точка-точка для передачи и приема при обмене данными между CPU и партнером.

CM включает данные из параметра BUFFER инструкции передачи в протокол 3964(R) и отправляет данные партнеру.

CM получает данные от партнера по протоколу 3964(R), удаляет информацию протокола и выводит данные в параметре BUFFER инструкции приема.

Дополнительную информацию можно найти в описаниях следующих инструкций точка-точка:

- Инструкция Send_P2P (отправить данные из передающего буфера) (Страница 1154)
- Инструкция Receive_P2P (разрешить прием сообщений) (Страница 1158)

Можно использовать и старые инструкции точка-точка для передачи и приема:

- SEND_PTP (отправить данные из передающего буфера) (Страница 1307)
- RCV_PTP (разрешить прием сообщений) (Страница 1309)

13.3.3 Конфигурирование свободно программируемой PtP коммуникации

При конфигурировании коммуникационных интерфейсов для свободно программируемой PtP коммуникации можно действовать следующим образом:

- Использование конфигурации устройства в STEP 7, чтобы задать параметры порта (скорость и четность), параметры передачи и приема. CPU сохраняет настройки в конфигурации устройства и применяет эти настройки после перезапуска и перехода из RUN в STOP.
- Установка параметров через инструкции Port_Config (Страница 1141), Send_Config (Страница 1144) и Receive_Config (Страница 1147). Настройки порта, установленные инструкциями, действительны, пока CPU находится в режиме RUN. Настройки порта возвращаются к настройкам из конфигурации устройства после перехода в STOP или перезапуска.

После конфигурирования аппаратных устройств (Страница 151), выполняется параметрирование коммуникационных интерфейсов путем выбора одного из CM в стойке или CB, если они были сконфигурированы.





Вкладка "Свойства" окна инспектора показывает параметры выбранного CM или CB. Выбрать вариант "Конфигурация порта" для редактирования следующих параметров:

- Скорость передачи
- Четность
- Число битов на символ
- Количество стоповых битов
- Управление потоком (только RS232)
- Время ожидания

Для CM 1241 RS232 и для CB RS485 (за исключением управления потоком (Страница 1125), который поддерживается только CM 1241 RS232) параметры конфигурации порта у коммуникационных модулей RS232 и RS485 и коммуникационный платы RS485 совпадают. Значения параметров могут отличаться.

Для CM 1241 RS422/485 предлагаются дополнительные опции для конфигурации порта, которые будет представлены ниже. Режим 422 модуля CM 1241 RS422/485 поддерживает и программное управление потоком.



Выбрать вариант "Конфигурация порта" для редактирования следующих параметров RS422/485:

- "Рабочий режим":
  - Дуплексный режим (RS422), четырехпроводный режим (соединение точка-точка)
  - Дуплексный режим (RS422), четырехпроводный режим (многоточечный ведущий)
  - Дуплексный режим (RS422), четырехпроводный режим (многоточечный ведомый)
  - Полудуплексный режим (RS485), двухпроводный режим
- "Начальное состояние линии приема":
  - Нет
  - Прямое смещение (сигнал R(A) 0 В, сигнал R(B) 5 В)

Программа пользователя STEP 7 также может сконфигурировать порт или изменить существующую конфигурацию с помощью инструкции Port_Config (Страница 1141). Раздел описания инструкции содержит детальную информацию о рабочем режиме и начальном состоянии линии, а также о других параметрах.

Параметр	Определение
Скорость передачи	Значение по умолчанию для скорости передачи составляет 9,6 кбит/с. Возможные значения: 300 бод, 600 бод, 1,2 кбит/с, 2,4 кбит/с, 4,8 кбит/с, 9,6 кбит/с, 19,2 кбит/с, 38,4 кбит/с, 57,6 кбит/с, 76,8 кбит/с и 115,2 кбит/с.
Четность	Установка по умолчанию для четности "без контроля четности". Возможные значения: без контроля четности, совпадение при контроле по чётности, совпадение при контроле по нечётности, контроль по единичному биту (бит четности всегда установлен в 1) и контроль по нулевому биту (бит четности всегда установлен в 0).
Число битов на символ	Количество битов данных в одном символе. Правильные установки 7 или 8.
Количество стоповых битов	Число стоповых битов может быть равно одному или двум. По умолчанию используется один.
Управление потоком	Для коммуникационного модуля RS232 можно выбрать аппаратное или программное управление потоком (Страница 1125). Если выбирается аппаратное управление потоком, то можно выбирать, включен ли RTS-сигнал всегда, или он переключается. Если выбирается программное управление потоком, то можно определить символы XON и XOFF. Коммуникационные интерфейсы RS485 не поддерживают управление потоком. Режим 422 модуля CM 1241 RS422/485 поддерживает программное управление потоком.
Время ожидания	Время ожидания определяет время, когда CM или CB ожидают получения CTS после передачи RTS, или для приема XON после приема XOFF, в зависимости от типа управления потоком. Если время ожидания истекло, прежде чем коммуникационный интерфейс получает ожидаемый CTS или XON, CM или CB отменяют операцию передачи и в программе пользователя появляется сообщение об ошибке. Время ожидания указывается в миллисекундах. Допустимый диапазон значений от 0 до 65535 миллисекунд.

Параметр	Определение
Рабочее состояние	Здесь выбирается рабочий режим RS422 или RS485 и конфигурации сети.
Начальное состояние линии приема	Здесь выбирается смещение. Допустимые значения: без смещения, прямое смещение и обратное смещение. Обратное смещение позволяет обнаружить обрыва кабеля.

13.3.3.1 Организация управления потоком

Управление потоком относится к механизму для балансировки передачи и приема данных с целью исключения их потери. Управление потоком гарантирует, что передающее устройство не отправит больше информации, чем может обработать принимающее устройство. Управление потоком возможно на аппаратном или программном уровне. Коммуникационный модуль RS232 поддерживает аппаратное и программное управление потоком. RS485 CM и CB не поддерживают управление потоком. Режим 422 модуля CM 1241 RS422/485 поддерживает программное управление потоком. Тип управления потоком указывается либо при настройке порта (Страница 1123), либо с помощью инструкции PORT_CFG (Страница 1298).

Аппаратное управление потоком работает через коммуникационные сигналы Request-to-send (RTS) и Clear-to-send (CTS). У коммуникационного модуля RS232 сигнал RTS снимается с контакта 7, а сигнал CTS с контакта 8. Коммуникационный модуль RS232 - это устройство DTE (оконечное оборудование данных), которое устанавливает RTS в качестве выхода и контролирует CTS в качестве входа.

Аппаратное управление потоком: RTS с переключением

Если включено аппаратное управление потоком с RTS-переключением коммуникационного модуля RS232, модуль устанавливает сигнал RTS на 1 для отправки данных. Он контролирует сигнал CTS, чтобы определить, готово ли принимающее устройство к приему данных. Если сигнал CTS активен, то модуль может передавать данные до тех пор, пока этот сигнал остается активным. Если сигнал CTS становится неактивным, то передача данных должна быть приостановлена.

Она будет возобновлена при повторном включении сигнала CTS. Если сигнал CTS не включится снова в течение установленного времени ожидания, то модуль отменяет передачу и возвращает ошибку в программу пользователя. Время ожидания должно быть указано в конфигурации порта (Страница 1123).

Управление потоком RTS с переключением полезно для устройств, которым требуется сигнал об активной передаче. Примером может служить радиомодем, который использует RTS в качестве "ключевого" сигнала для активации радиопередатчика. Управление потоком RTS с переключением не будет работать со стандартными телефонными модемами. Для телефонных модемов следует использовать вариант "RTS всегда включен".

Аппаратное управление потоком: RTS всегда включен

В режиме "RTS всегда включен", CM 1241 устанавливает RTS активным по умолчанию. Устройство, напр., телефонный модем, контролирует сигнал RTS от CM и использует этот сигнал как сообщение о готовности к передаче. Модем выполняет передачу на CM только тогда, когда RTS активен, то есть когда телефонный модем видит активный CTS. Если RTS неактивен, то телефонный модем не передает данные в CM.

Чтобы разрешить модему отправлять данные в CM в любое время, установить для аппаратного управления потоком "RTS всегда включен". В этом случае CM всегда устанавливает сигнал RTS на 1. CM не будет отключать RTS и тогда, когда модуль не может принимать символы. Передающее устройство должно гарантировать, что оно не станет причиной переполнения приемного буфера CM.

Сигналы Data Terminal Ready (DTR) и Data Set Ready (DSR)

CM устанавливает DTR активным для любого типа аппаратного управления потоком. Модуль выполняет передачу только тогда, когда сигнал DSR становится активным. Состояние DSR оценивается только в начале операции передачи. Если DSR становится неактивным после начала передачи, передача не будет приостановлена.

Программное управление потоком

Для программного управления потоком используются специальные символы в сообщениях. Можно конфигурировать шестнадцатеричные символы, представляющие XON и XOFF.

XOFF указывает, что передача должна быть остановлена. XON указывает, что передача может возобновиться. Символы XOFF и XON не должны совпадать.

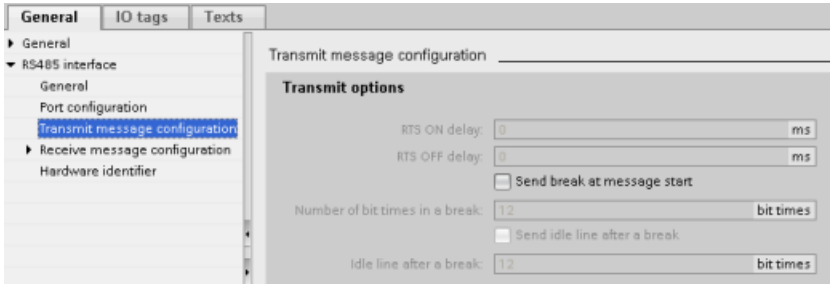
Когда передающее устройство получает символ XOFF от принимающего устройства, оно прекращает передачу. Передача возобновляется, когда передающее устройство получает символ XON. Если CM не получает символ XON в течение времени ожидания, указанного в конфигурации порта (Страница 1123), CM прерывает передачу и возвращает ошибку программе пользователя.

Программное управление потоком требует дуплексной связи, так как принимающий партнер должен иметь возможность отправлять XOFF передающему партнеру во время передачи. Программное управление потоком возможно только с сообщениями, которые содержат только символы ASCII. Двоичные протоколы не могут использовать программное управление потоком.

Прежде чем будет установлена свободно программируемая коммуникация точка-точка, необходимо сконфигурировать параметры для передачи и приема сообщений. Эти параметры определяют, как работает коммуникация, когда сообщения передаются на / принимаются от целевого устройства.

13.3.3.2 Конфигурирование параметров передачи

В конфигурации CPU настраивается, как интерфейс связи передает данные. Для этого устанавливаются свойства "Конфигурация передачи сообщения" для выбранного интерфейса.



Также можно динамически конфигурировать или изменять параметры для передачи сообщения из пользовательской программы, используя инструкцию Send_Config (Страница 1144).

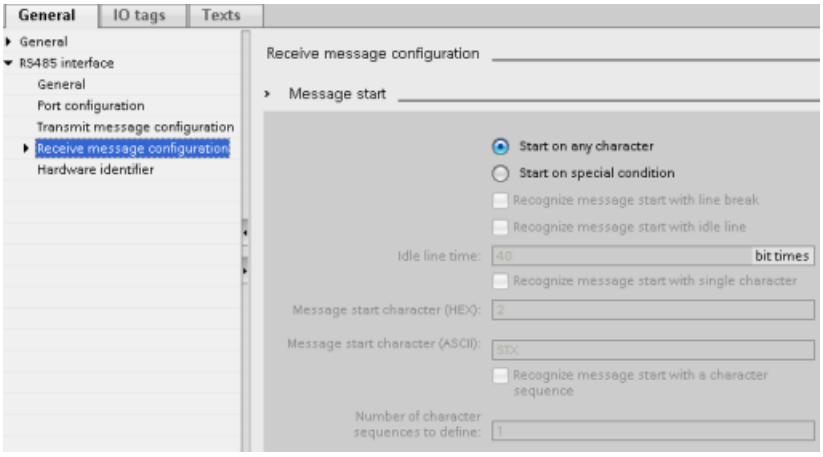
Примечание

Значения параметров, установленные инструкцией Send_Config в пользовательской программе, переопределяют свойства "Конфигурация передачи сообщения". Параметры, установленные из инструкции Send_Config, в случае отключения питания не сохраняются в CPU.

Параметр	Определение
Задержка включения RTS	Время ожидания от включения RTS до начала передачи. Действительный диапазон составляет от 0 до 65535 мс, предустановка 0. Данный параметр действителен только при установленном в конфигурации порта (Страница 1123) аппаратном контроле потока. CTS обрабатывается после истечения времени задержки включения RTS. Этот параметр применим только для модулей RS232.
Задержка выключения RTS	Время ожидания до отключения RTS после завершения передачи. Действительный диапазон составляет от 0 до 65535 мс, предустановка 0. Данный параметр действителен только при установленном в конфигурации порта (Страница 1123) аппаратном контроле потока. Этот параметр применим только для модулей RS232.
Пауза передачи в начале сообщения Число битовых интервалов за паузу	Указывает, что после начала сообщения будет отправляться пауза после истечения задержки включения RTS (если настроено) и при активности CTS. Можно установить, сколько битовых интервалов будет длиться пауза, при которой передача не выполняется. По умолчанию установлено значение 12, а максимальное значение равно 65535, но не более восьми секунд.
Передать информацию о простое линии после паузы Простой линии после паузы	Определяет, что сигнал свободной линии будет отправлен перед началом сообщения. Он отправляется после паузы, если таковая сконфигурирована. Параметр "Простой линии после паузы" определяет, сколько битовых интервалов содержит сигнал свободной линии. По умолчанию установлено значение 12, а максимальное значение равно 65535, но не более восьми секунд.

13.3.3.3 Конфигурирование параметров приема

В конфигурации устройства определяется, как интерфейс связи должен принимать данные и как он распознает начало и конец сообщения. Эти параметры задаются для выбранного интерфейса в свойствах "Конфигурация приема сообщения".



Также можно динамически конфигурировать или изменять параметры для сообщения из пользовательской программы, используя инструкцию Receive_Config (Страница 1147).

Примечание

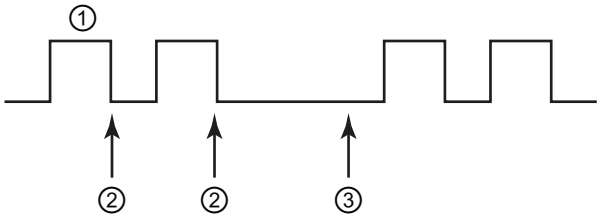
Значения параметров, установленные Receive_Config в пользовательской программе, переопределяют свойства "Конфигурация приема сообщения". Параметры, установленные из инструкции RCV_CFG, в случае отключения или перехода в STOP не сохраняются в CPU.

Условия для начала сообщения

Можно определить, как коммуникационный интерфейс распознает начало сообщения. Стартовые символы и содержимое сообщения помещается в принимающий буфер, пока не будет выполнено сконфигурированное условие окончания.

Может быть задано несколько условий начала сообщения. Если указаны несколько условий начала, для запуска передачи данных все условия должны быть соблюдены. Например, если сконфигурировать время для простоя линии и определенный начальный символ, то CM или CB будут сначала ожидать соблюдения требования свободной линии, а затем CM будет искать указанный начальный символ. Если будет принят какой-либо другой символ (не являющийся указанным начальным символом), то CM или CB перезапустят поиск начала сообщения, с учетом времени ожидания свободной линии.

Параметр	Определение
Начало с любого символа	Это условие указывает, что сообщение начинается при успешном приеме любого символа. Этот символ будет первым символом в сообщении.
Пауза на линии	Это условие указывает, что операция по приему сообщения должна начаться после приема символа паузы.

Параметр	Определение
Линия на паузе	Это условие указывает, что прием сообщения должен начаться после того, как принимающая линия в течение заданного числа битовых интервалов находилась в простое. Как только это условие будет выполнено, начнется прием сообщения.  ① Символы ② Перезапуск таймера простоя линии ③ Свободная линия обнаружена, и запущен прием сообщения
Специальное условие: Распознавать начало сообщения по одному символу	Указывает, что признаком начала сообщения является определенный символ. Этот символ будет первым символом в сообщении. Любой символ, который будет получен перед этим определенным символом, игнорируется. Символом по умолчанию является STX.
Специальное условие: Распознавать начало сообщения по последовательности символов	Указывает, что признаком начала сообщения является последовательность определенных символов, состоящая макс. из четырех сконфигурированных рядов. Для каждой последовательности можно определить до пяти символов. Для позиции каждого символа можно указать конкретный шестнадцатеричный символ, или что этот символ игнорируется при сопоставлении последовательностей (подстановочный символ). Последний определенный символ последовательности завершает эту последовательность условия начала сообщения. Входящие последовательности сравниваются со сконфигурированными условиями начала сообщения, пока какое-либо условие начала не будет выполнено. Как только условие начала сообщения выполнено, начинается анализ условия конца сообщения. Можно сконфигурировать до четырех определенных символьных последовательностей. Используется несколько последовательностей условия начала, когда различные последовательности символов могут указать на начало сообщения. Если какая-либо из символьных последовательностей совпадает, то сообщение начинается.

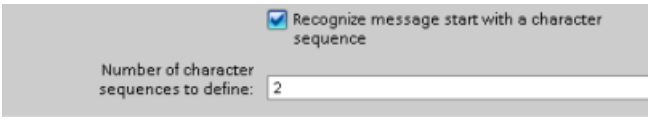
Условия начала проверяются в следующей последовательности:

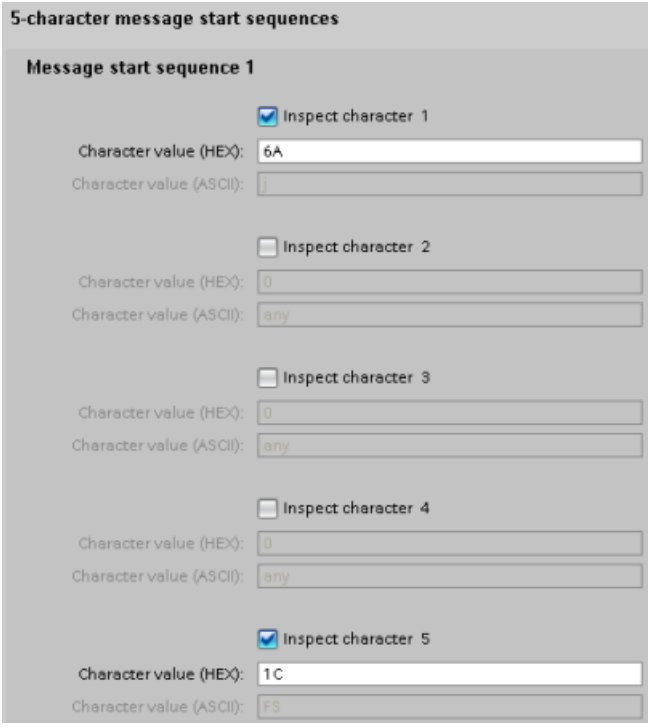
- Простой линии
- Разрыв линии
- Символы или последовательности символов

При проверке на несколько условий начала, если одно из условий не соблюдено, CM или CB перезапустят проверку, начиная с первого требуемого условия. После того, как CM или CB устанавливает, что условия начала соблюдены, начинается анализ условия окончания.

Пример конфигурации: Начинать сообщение по одной из двух последовательностей символов

Рассмотрим следующую конфигурацию условий для начала сообщения:







В этой конфигурации условие начала сообщения выполнено, если имеет место одна из следующих комбинаций символов:

- Когда принимается последовательность из пяти символов, в которой первым символом является 0x6A, а пятым 0x1C. В этой конфигурации в позициях 2, 3 и 4 могут находиться любые символы. После приема пятого символа начинается мониторинг условий конца сообщения.
- Когда принимаются последовательно два символа 0x6A, которым предшествует любой другой символ. В этом случае мониторинг условий конца сообщения начинается после приема второй последовательности 0x6A (3 символа). Символ, предшествующий первому символу 0x6A, включается в условие начала сообщения.

Примеры последовательностей, удовлетворяющих условиям начала сообщения:

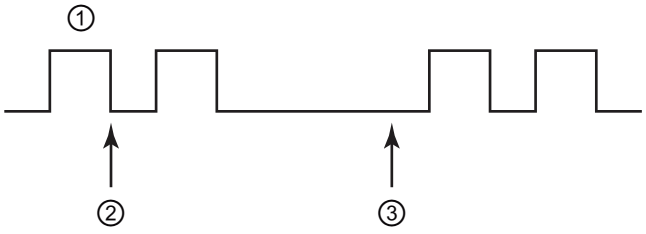
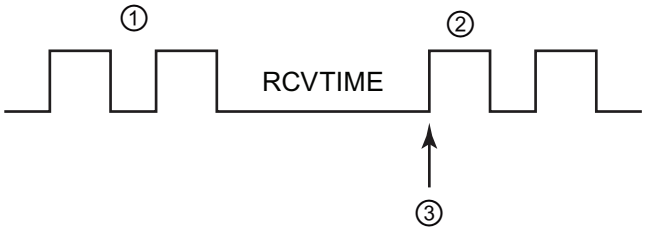
- <любой символ> 6A 6A
- 6A 12 14 18 1C
- 6A 44 A5 D2 1C

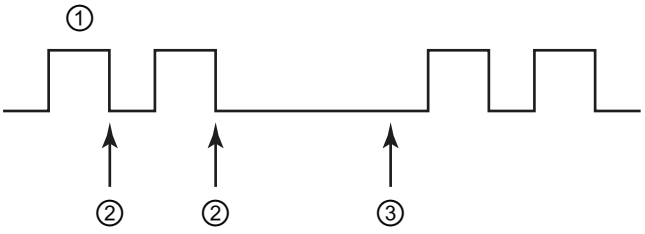
Условия для конца сообщения

Также можно определить, как интерфейс связи должен распознавать конец сообщения. Можно сконфигурировать несколько условий окончания сообщения. Если совпадает одно из этих условий, сообщение заканчивается.

Например, можно задать условие конца сообщения с тайм-аутом сообщения в 300 мс, временем ожидания очередного символа в течение 40 битовых интервалов и максимальной длиной 50 байт. Сообщение завершается, если его прием займет больше

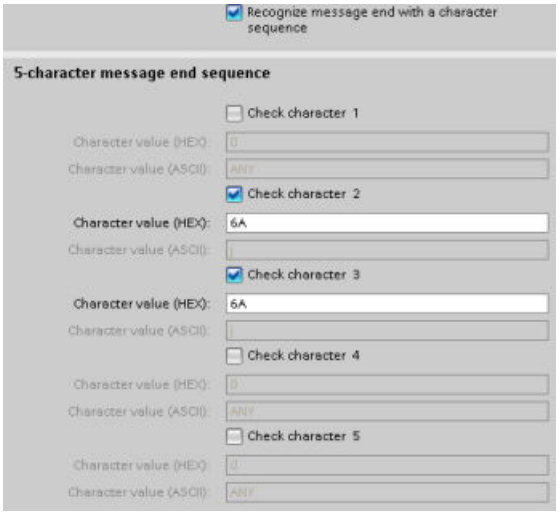
300 миллисекунд, или если промежуток между двумя любыми символами превышает 40 битовых интервалов, или если было принято 50 байтов.

Параметр	Определение
Распознавание конца сообщения по тайм-ауту сообщения	Конец сообщения достигнут, когда истекает сконфигурированный интервал времени ожидания конца сообщения. Отсчет тайм-аута сообщения начинается, когда было удовлетворено условие начала сообщения. Значение по умолчанию равно 200 мс, а диапазон составляет от 0 до 65535 мс.  ① Принятые символы ② Условие для начала сообщения выполнено: таймер сообщения запускается ③ Таймер сообщения истек и прерывает сообщение
Распознавание конца сообщения по тайм-ауту ответа	Конец сообщения достигнут, когда сконфигурированное время ожидания ответа истекло, прежде чем принята допустимая последовательность начала. Отсчет тайм-аута для ответа начинается, когда передача завершается, и CM или CB запускает инструкцию приема. Значение по умолчанию для тайм-аута ответа равно 200 мс, а диапазон составляет от 0 до 65535 мс. Если в течение указанного RCVTIME периода времени для ответа символ не был получен, то ошибка возвращается соответствующей инструкции RCV_PTP. Тайм-аут ответа не определяет специфическое условие окончания. Он определяет только то, что символ должен быть успешно принят в течение требуемого времени. Необходимо сконфигурировать другое условие окончания, чтобы указать на фактический конец сообщения.  ① Переданные символы ② Принятые символы ③ Первый символ должен быть успешно принят к этому моменту.

Параметр	Определение
Распознавать конец сообщения по промежутку между символами	Конец сообщения достигнут, когда истекло максимальное сконфигурированное время между любыми двумя последовательными символами сообщения. Значение по умолчанию для межсимвольного интервала составляет 12, а максимальное количество - 65535 битовых интервалов, до максимума равного 8 секундам.  ① Принятые символы ② Перезапуск таймера для времени между двумя символами ③ Таймер для времени между двумя символами истекает и завершает сообщение.
Распознавать конец сообщения по получению фиксированного числа символов	Конец сообщения достигнут, если было получено сконфигурированное число символов. Действительный диапазон для фиксированной длины лежит между 1 и 4096. Следует учитывать, что это условие для конца сообщения действует для S7-1200 только от CPU версии V4.0.
Распознавать конец сообщения по максимальной длине сообщения	Конец сообщения достигнут, если было получено сконфигурированное максимальное число символов. Действительный диапазон для максимальной длины лежит между 1 и 1024. Этой условие может использоваться для исключения ошибки "Переполнение буфера сообщений". Когда это условие конца объединено с условиями тайм-аута окончания, и возникает условие тайм-аута окончания, любые допустимые полученные символы выводятся, даже если максимальная длина не была достигнута. Это обеспечивает поддержку для протоколов различной длины, когда известна только максимальная длина.
Чтение длины сообщения из сообщения	Сведения о длине сообщения встроены в само сообщение. Конец сообщения достигнут, если было получено сообщение с указанной длиной. Метод для определения и интерпретации длины сообщения описан ниже.
Распознавать конец сообщения по одному символу	Конец сообщения достигнут, если был получен определенный символ.
Распознавать конец сообщения по последовательности символов	Конец сообщения достигнут, если была получена определенная последовательность символов. Может быть указана последовательность с длиной до 5 символов. Для позиции каждого символа можно указать конкретный шестнадцатеричный символ, или что этот символ должен игнорироваться в последовательности. Вводные игнорируемые символы не включаются в условие окончания. Заключительные игнорируемые символы включаются в условие окончания.

Пример конфигурации: Завершение сообщения по последовательности символов

Рассмотрим следующую конфигурацию условий для конца сообщения:



В этом случае условие завершения удовлетворено, когда приняты два последовательных символа 0x6A, сопровождаемые любыми двумя символами. Символ, предшествующий 0x6A 0x6A, не является частью символьной последовательности завершения. После 0x6A 0x6A требуются два символа, чтобы дополнить символьную последовательность завершения. Значения, принятые в символьных позициях 4 и 5, не важны, но они должны быть приняты, чтобы удовлетворить условию завершения.

Примечание

Если символьная последовательность должна указать на конец сообщения, поместить ее на последние знаковые позиции. В примере выше, для того, чтобы 0x6A 0x6A завершали сообщение без заключительных символов, следует сконфигурировать 0x6A на символьных позициях 4 и 5.

Указание длины сообщения в сообщении

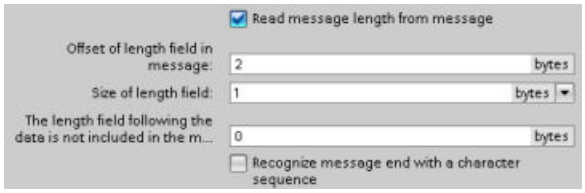
При выборе специального условия, когда длина сообщения включена в сообщение, необходимо указать три параметра, которые содержат информацию о длине сообщения.

Фактическая структура сообщения зависит от используемого протокола. Этими тремя параметрами являются:

- n: Позиция символа (по отношению к 1) в сообщении, с которой начинается спецификатор длины
- Размер спецификатора длины: Число байт (один, два или четыре) в спецификаторе длины
- Длина m: Число символов после спецификатора длины, которые не учитываются в расчете длины

Символы конца не должны быть последовательными. Значение "Длина m" может использоваться, чтобы определить длину поля контрольной суммы, размер которого не включен в поле длины.

Эти поля появляются в конфигурации приема сообщения в свойствах устройства:



Пример 1: Рассмотрим сообщение, структурированное в соответствии со следующим протоколом:

STX	Len (n)	Символы от 3 до 14, подсчитываются по длине											
		ADR	PKE		INDEX		PWD		STW		HSW		BCC
1	2	3	4	5	6	7	8	9	10	11	12	13	14
STX	0x0C	xx	xxxx		xxxx		xxxx		xxxx		xxxx		xx

Параметры для длины принимаемого сообщения конфигурируются следующим образом:

- n = 2 (длина сообщения начинается с байта 2)
- Размер спецификатора длины = 1 (длина сообщения определена в одном байте)
- Длина m = 0 (После спецификатора длины нет других символов, которые не учитываются при расчете длины. За указателем длины следуют 12 символов.)

В этом примере символы с 3 по 14 включительно подсчитываются с помощью Len (n).

Пример 2: Рассмотрим сообщение, структурированное в соответствии со следующим протоколом:

SD1	Len (n)	Len (n)	SD2	Символы от 5 до 10, подсчитываются по длине						FCS	ED
				DA	SA	FA	Элемент данных=3 байт				
1	2	3	4	5	6	7	8	9	10	11	12
xx	0x06	0x06	xx	xx	xx	xx	xx	xx	xx	xx	xx

Параметры для длины принимаемого сообщения конфигурируются следующим образом:

- n = 3 (длина сообщения начинается с байта 3)
- Размер спецификатора длины = 1 (длина сообщения определена в одном байте)
- Длина m = 3 (За спецификатором длины следует три символа, которые не учитываются при расчете длины.) В протоколе этого примера символы SD2, FCS и ED не рассматриваются при расчете длины. Другие шесть символов учитываются при подсчете длины; поэтому общее количество символов после спецификатора длины равняется девяти.)

В этом примере символы с 5 по 10 включительно подсчитываются с помощью Len (n).

13.3.4 Конфигурирование 3964(R) коммуникации

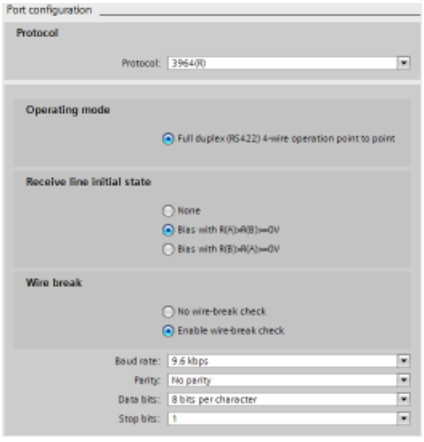
13.3.4.1 Конфигурирование 3964(R) коммуникационных портов

При конфигурировании коммуникационных интерфейсов для 3964(R) коммуникации можно действовать следующим образом:

- Конфигурирование параметров портов в конфигурации устройства в STEP 7. CPU сохраняет настройки в конфигурации устройства и применяет эти настройки после перезапуска.
- Установка параметров портов через инструкцию Port_Config (Страница 1141). Настройки порта, установленные инструкциями, действительны, пока CPU находится в режиме RUN. Настройки порта возвращаются к настройкам из конфигурации устройства после перезапуска.

После добавления коммуникационных интерфейсов в конфигурацию устройства (Страница 155), выполняется параметрирование коммуникационных интерфейсов путем выбора одного из CM в стойке.





Вкладка "Свойства" окна инспектора показывает параметры выбранного CM. Выбрать вариант "Конфигурация порта" для редактирования следующих параметров:

- Протокол: 3964(R)
- Рабочий режим (только модуль CM 1241 (RS-422/485))
- Рабочий режим (только модуль CM 1241 (RS-422/485))
- Обрыв кабедя (только модуль CM 1241 (RS-422/485))
- Скорость передачи
- Четность
- Биты данных
- Стоповые биты

Параметр	Определение
Протокол	3964R или свободное программирование. Выберите вариант 3964R, чтобы сконфигурировать порт для 3964(R)-коммуникации.
Рабочий режим*	Дуплекс (RS-422), 4-проводный режим, точка-точка. (активирован)
Начальное состояние линии приема*	Активировать одну из следующих возможностей: • Нет • Смещение с R(A)>R(B)>=0 В • Смещение с R(B)>R(A)>=0 В

Параметр	Определение
Обрыв провода*	Активировать одну из следующих возможностей: • Без контроля обрыва провода • Включить контроль обрыва провода
Скорость передачи	Значение по умолчанию для скорости передачи составляет 9,6 кбит/с. Возможные значения: 300 бод, 600 бод, 1,2 кбит/с, 2,4 кбит/с, 4,8 кбит/с, 9,6 кбит/с, 19,2 кбит/с, 38,4 кбит/с, 57,6 кбит/с, 76,8 кбит/с и 115,2 кбит/с.
Четность	Установка по умолчанию для четности "без контроля четности". Возможные значения: Без контроля четности, совпадение при контроле по чётности, совпадение при контроле по нечётности, контроль по единичному биту (бит четности всегда установлен в 1) и контроль по нулевому биту (бит четности всегда установлен в 0) и любая четность (установить бит четности для передачи на 0; игнорировать ошибки четности при приеме).
Число битов на символ	Количество битов данных в одном символе. Правильные установки 7 или 8.
Количество стоповых битов	Число стоповых битов может быть равно одному или двум. По умолчанию используется один.

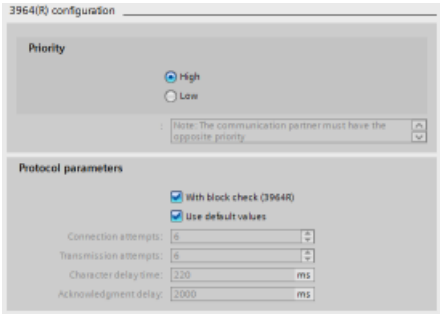
* Только модуль CM 1241 (RS-422/485)

13.3.4.2 Конфигурирование приоритета 3964(R) и параметров протокола

При конфигурировании коммуникационных интерфейсов для 3964(R) коммуникации можно действовать следующим образом:

- Кликнуть мышью в конфигурации устройства коммуникационного интерфейса по "Конфигурация 3964(R)", чтобы определить приоритет и сконфигурировать параметры протокола. CPU сохраняет настройки в конфигурации устройства и применяет эти настройки после перезапуска.
- Определить приоритет с помощью инструкции P3964_Config (Страница 1152) и сконфигурировать параметры протокола. Установленные инструкциями значения действительны, пока CPU находится в режиме RUN. Значения возвращаются к настройкам из конфигурации устройства после перезапуска.





Вкладка "Свойства" окна инспектора показывает параметры выбранного CM. Выбрать вариант "Конфигурация 3964(R)" для редактирования следующих параметров:

- Приоритет (высокий или низкий)
- Параметры протокола
  - С контролем блока (3964R)
  - Использовать стандартные значения
    Попытки соединения
    Попытки передачи
    Время задержки символа
    Задержка квитирования

Параметр	Определение
Приоритет	Высокий или низкий: CM имеет высокий или низкий приоритет, приоритет партнера должен быть противоположным.
С контролем блока (3964)	Если выбрано, то защита передачи при коммуникации 3964(R) обеспечивается путем включения символа контроля блока (BCC). Если эта опция не выбрана, то символ контроля блока не используется при безопасной передаче данных.
Использовать значения по умолчанию	Если выбрано, то 3964(R) использует значения по умолчанию для следующих параметров протокола: • Попытки соединения • Попытки передачи • Время задержки символа • Задержка квитирования Если опция не выбрана, то для каждого из этих параметров могут быть сконфигурированы значения.
Попытки соединения	Количество попыток создания соединения (по умолчанию: 6 попыток соединения) 1 до 255
Попытки передачи	Количество попыток передачи данных (по умолчанию: 6 попыток соединения) 1 до 255
Время задержки символа	Настройка времени задержки символа (зависит от настроенной скорости передачи данных) (значение по умолчанию: 220 мс) от 1 до 65535 мс
Задержка квитирования	Настройка времени задержки квитирования (зависит от настроенной скорости передачи данных) (значение по умолчанию: 2000 мс, если активирован контроль блока; 550 мс, если не активирован контроль блока) от 1 мс до 65535 мс

Примечание

За исключением приоритета, прочие настройки протокола у CM модуля и партнера должны совпадать.

13.3.5 Инструкции точка-точка

13.3.5.1 Общие параметры для операций точка-точка

Таблица 13-3 Общие входные параметры инструкций PTP

Параметр	Описание
REQ	Многие из PtP инструкции имеют вход REQ, который инициирует инструкцию по положительному фронту (0 на 1). Вход REQ должен быть 1 (TRUE) при выполнении инструкции, но вход REQ может оставаться TRUE столько времени, сколько необходимо. Инструкция не инициирует другую операцию, пока она не будет вызвана с входом REQ = FALSE, чтобы команда могла сбросить состояние входа REQ. Это необходимо для того, чтобы инструкция могла обнаружить положительный фронт для запуска следующей операции. В случае добавления в программу PtP инструкции, STEP 7 потребует указать экземплярный блок данных. Для каждого вызова инструкции PtP следует использовать уникальный блок данных. Это гарантирует, что такие входы как REQ будут правильно обрабатываться каждой инструкцией.
PORT	Адрес порта присваивается во время конфигурации коммуникационного модуля. После конфигурации для стандартного порта можно выбрать символьное имя из выпадающего списка параметра. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Константы" таблицы переменных PLC.
Разрешение битового интервала	Для разных параметров при сконфигурированной скорости передачи данных указывается количество битовых интервалов. При указании параметра в битовых интервалах он не зависит от скорости передачи данных. Для всех параметров в единицах битового интервала может быть указано максимальное количество в 65535. Однако максимальный промежуток времени, замеряемый CM или CB, составляет 8 с.

Выходные параметры DONE, NDR, ERROR и STATUS инструкций PtP отображают статус выполнения функций PtP.

Таблица 13-4 Выходные параметры DONE, NDR, ERROR и STATUS

Параметр	Тип данных	По умолчанию	Описание
DONE	Bool	FALSE	Устанавливается на TRUE на один цикл выполнения, чтобы показать, что последний запрос был завершен с ошибками; в противном случае FALSE.
NDR	Bool	FALSE	Устанавливается на TRUE на один цикл выполнения, чтобы показать, что запрошенная операция была завершена без ошибок и были получены новые данные; в противном случае FALSE.

Параметр	Тип данных	По умолчанию	Описание
ERROR	Bool	FALSE	Устанавливается для одного цикла выполнения на TRUE, чтобы показать, что последний запрос завершен с ошибками, соответствующий код ошибки находится в STATUS; в противном случае FALSE.
STATUS	Word	0	Статус результата: • Если устанавливается бит DONE или NDR, то STATUS устанавливается на 0 или на информационный код. • Если устанавливается бит ERROR, то STATUS устанавливается на код ошибки. • Если не устанавливается ни один из вышеуказанных битов, инструкция может вернуть результаты состояния, которые описывают текущее состояние функции. Значение STATUS сохраняется при выполнении функции.

Примечание

Проследить за тем, чтобы параметры DONE, NDR и ERROR были установлены только для одного цикла. Программная логика должна сохранить временные значения состояния выхода в ячейках данных, чтобы можно было обнаружить изменения состояния в последующих циклах программы.

Таблица 13-5 Общие коды условий

STATUS (W#16#....)	Описание
0000	Ошибки отсутствуют
7000	Функция не занята
7001	Функция занята первым вызовом.
7002	Функция занята последующими вызовами (опросы после первого вызова).
8x3A	Недопустимый указатель в параметре x
8070	Вся внутренняя память используется, слишком многих параллельно обрабатываемых инструкций
8080	Недопустимый номер порта.
8081	Превышение времени, ошибка модуля или другая внутренняя ошибка
8082	Не удалось выполнить параметрирование, т.е. параметрирование выполняется в фоновом режиме.
8083	Переполнение буфера: CM или CB возвратили полученное сообщение с длиной, больше допустимой.
8090	Внутренняя ошибка: Неверная длина сообщения, неправильный субмодуль или недопустимое сообщение Обратиться в службу сервиса для клиентов.
8091	Внутренняя ошибка: Неправильная версия в сообщении параметризации Обратиться в службу сервиса для клиентов.
8092	Внутренняя ошибка: Неверная длина записи в сообщении параметризации Обратиться в службу сервиса для клиентов.

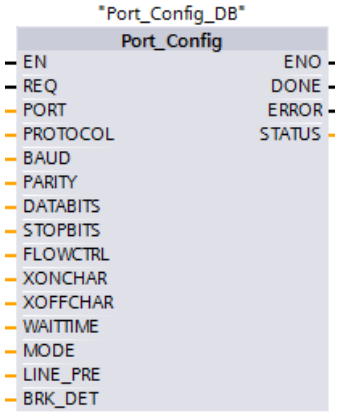
Таблица 13-6 Общие классы ошибок

Описание класса	Классы ошибок	Описание
Конфигурация порта	16#81Ax	Для описания частых ошибок в конфигурации интерфейса
Конфигурация передачи	16#81Bx	Для описания частых ошибок в конфигурации передачи
Конфигурация приема	16#81Cx 16#82Cx	Для описания частых ошибок в конфигурации приема
Время выполнения передачи	16#81Dx	Для описания частых ошибок в процессе передачи
Время выполнения приема	16#81Ex	Для описания частых ошибок в процессе приема
Обработка сигнала	16#81Fx	Для описания частых ошибок, связанных с обработкой сигналов
Ошибка указателя	от 16#8p01 до 16#8p51	Для ошибок указателя ANY, где "p" - номер параметра инструкции
Ошибки встроенного протокола	16#848x 16#858x	Для ошибок встроенного протокола

13.3.5.2 Инструкция Port_Config (динамическое конфигурирование коммуникационных параметров)

Таблица 13-7 Инструкция Port_Config (конфигурирование порта)

LAD/FBD	SCL	Описание

	`"Port_Config_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_word_in_,`
	`PROTOCOL:=_uint_in_,`
	`BAUD:=_uint_in_,`
	`PARITY:=_uint_in_,`
	`DATABITS:=_uint_in_,`
	`STOPBITS:=_uint_in_,`
	`FLOWCTRL:=_uint_in_,`
	`XONCHAR:=_char_in_,`
	`XOFFCHAR:=_char_in_,`
	`WAITTIME:=_uint_in_,`
	`MODE:=_uint_in_,`
	`LINE_PRE:=_uint_in_,`
	`BRK_DET:=_uint_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		Инструкция Port_Config позволяет изменять параметры порта, такие как скорость обмена, из программы.
		Можно настроить начальную статическую конфигурацию порта в свойствах конфигурации устройства или просто использовать значения по умолчанию. После с помощью инструкции Port_Config можно изменять конфигурацию в программе.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

CPU не хранит постоянно значения, которые определяются с помощью инструкции Port_Config. Параметры, сохраненные в конфигурации устройства, восстанавливаются CPU при переходе из RUN в STOP или при повторном включении. Дополнительную

информацию можно найти здесь: Конфигурирование коммуникационных портов (Страница 1123) и Организация управления потоком (Страница 1125).

Таблица 13-8 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Активирует изменение конфигурации при положительном фронте данного входа. (значение по умолчанию: False)
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC. (значение по умолчанию: 0)
PROTOCOL	IN	UInt	0 - Свободно программируемая коммуникация (значение по умолчанию) 1. Протокол 3964(R)
BAUD	IN	UInt	Скорость передачи порта (значение по умолчанию: 6): 1 = 300 бод, 2 = 600 бод, 3 = 1200 бод, 4 = 2400 бод, 5 = 4800 бод, 6 = 9600 бод, 7 = 19200 бод, 8 = 38400 бод, 9 = 57600 бод, 10 = 76800 бод, 11 = 115200 бод
PARITY	IN	UInt	Четность порта (значение по умолчанию: 1): 1 = без контроля четности, 2 = совпадение при контроле по чётности, 3 = совпадение при контроле по нечётности, 4 = контроль по единичному биту, 5 = контроль по нулевому биту
DATABITS	IN	UInt	Битов на символ (значение по умолчанию: 1): 1 = 8 битов данных, 2 = 7 битов данных
STOPBITS	IN	UInt	Стоповые биты (значение по умолчанию: 1): 1 = 1 стоповый бит, 2 = 2 стоповых бита
FLOWCTRL*	IN	UInt	Управление потоком (значение по умолчанию: 1): 1 = без управления потоком, 2 = XON/XOFF, 3 = аппаратный RTS всегда ВКЛ, 4 = аппаратный RTS с переключением
XONCHAR[1]	IN	Char	Задает символ, который будет использоваться в качестве символа XON. Обычно это символ DC1 (16#11). Данный параметр анализируется только при активированном управлении потоком. (значение по умолчанию: 16#11)
XOFFCHAR[1]	IN	Char	Задает символ, который будет использоваться в качестве символа XOFF. Обычно это символ DC3 (16#13). Данный параметр анализируется только при активированном управлении потоком. (значение по умолчанию: 16#13)
WAITTIME[1]	IN	UInt	Определяет время ожидания символа XON после получения символа XOFF, или время ожидания сигнала CTS после активации RTS (от 0 до 65535 мс). Данный параметр анализируется только при активированном управлении потоком. (значение по умолчанию: 2000)

Параметр и тип данных		Тип данных	Описание
MODE[2]	IN	UInt	Определяет выбор рабочего режима модуля. • 0 = дуплекс (RS232) • 1 = дуплекс (RS-422), 4-проводный режим, точка-точка, передатчик всегда включен • 2 = дуплекс (RS-422), 4-проводный режим (многоточечный ведущий), передатчик всегда включен • 3 = дуплекс (RS-422), 4-проводный режим (многоточечный ведомый), передатчик включен при отправке • 4 = полудуплекс (RS-485), двухпроводной режим
LINE_PRE	IN	UInt	Определяет неактивное состояние линии (простой). Для модулей RS422 и RS485 состояние свободной линии устанавливается приложением напряжение смещения к сигналам R(A) и R(B). Возможны следующие настройки: • 0 = без смещения (без предварительных установок) (по умолчанию) • 1 = смещение R(A) > R(B) ≥ 0B; только RS-422 • 2 = смещение R(B) > R(A) ≥ 0B; RS-422 и RS-485
BRK_DET	IN	UInt	Включает/выключает обнаружение обрыва коммуникационного кабеля. При включенном обнаружении разрыва кабеля, модуль сигнализирует ошибку, если коммуникационный кабель не подключен к модулю. В режиме RS422 "точка-точка" обнаружение обрыва кабеля возможно, только когда используется предварительная установка приемной линии со смещением, так что R (A) > R (B) ≥ 0B. • 0 = без обнаружения обрыва кабеля (по умолчанию) • 1 = обнаружение обрыва кабеля включено
DONE	OUT	Bool	Для целого цикла выполнения TRUE, после того как последний запрос был выполнен без ошибок.
ERROR	OUT	Bool	Для целого цикла выполнения TRUE, после того как последний запрос был выполнен с ошибками.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

[1] Не действует, если протокол = 1 (протокол 3964(R))

[2] Действительны только режимы работы 0 и 1, если протокол = 1 (протокол 3964(R)). Это зависит от того, является ли модулем CM модуль RS-232 или модуль RS-422.

Таблица 13-9 Коды условий

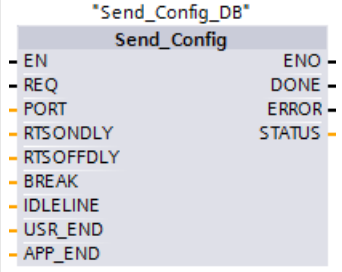
STATUS (W#16#....)	Описание
81A0	Указанный протокол не существует.
81A1	Указанная скорость передачи не существует.
81A2	Указанная опция четности не существует.
81A3	Указанное число битов данных не существует.
81A4	Указанное число стоповых битов не существует.
80A5	Указанный вариант управления потоком не существует.
81A6	Время ожидания 0 и управление потоком активировано
81A7	XON и XOFF - это недопустимые значения (например, одинаковое значение)

STATUS (W#16#....)	Описание
81A8	Ошибка в заголовке блока (например, неправильный тип блока или неправильная длина блока)
81A9	Реконфигурирование отклонено, так как выполняется конфигурирование
81AA	Недопустимый режим работы RS422/RS485
81AB	Недопустимая предварительная установка приемной линии для обнаружения обрыва
81AC	Недействительная обработка паузы RS232
8280	Отрицательное подтверждение при чтении из модуля
8281	Отрицательное подтверждение при записи в модуль
8282	Ведомое устройство DP или модуль недоступны

13.3.5.3 Инструкция Send_Config (динамическое конфигурирование параметров для последовательной коммуникации)

Таблица 13-10 Инструкция Send_Config (конфигурация передачи)

LAD/FBD	SCL	Описание

	`"Send_Config_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_word_in_,`
	`RTSONDLY:=_uint_in_,`
	`RTSOFFDLY:=_uint_in_,`
	`BREAK:=_uint_in_,`
	`IDLELINE:=_uint_in_,`
	`USR_END:=_string_in_,`
	`APP_END:=_string_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		Инструкция Send_Config обеспечивает динамическое конфигурирование параметров последовательной передачи для PtP коммуникационного порта. Любые сообщения в очереди в CM или CB сбрасываются, когда выполняется Send_Config.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Можно настроить начальную статическую конфигурацию порта в свойствах конфигурации устройства или просто использовать значения по умолчанию. После с помощью инструкции Send_Config можно изменять конфигурацию в программе.

CPU не хранит постоянно значения, которые определяются с помощью инструкции Send_Config. Параметры, сохраненные в конфигурации устройства, восстанавливаются

CPU при переходе из RUN в STOP или при повторном включении. См. Конфигурирование параметров передачи (Страница 1127)

Таблица 13-11 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Активирует изменение конфигурации при положительном фронте данного входа. (значение по умолчанию: False)
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC. (значение по умолчанию: 0)
RTSONDLY	IN	UInt	Количество миллисекунд, которые должны пройти после активации RTS, прежде чем будет выполнена передача данных Tx. Данный параметр действителен только при активированном аппаратном управлении потоком. Действительный диапазон находится между 0 и 65535 мс. Значение 0 выключает функцию. (значение по умолчанию: 0)
RTSOFFDLY	IN	UInt	Количество миллисекунд, которые должны пройти после передачи данных Tx, прежде чем будет выключен RTS: Данный параметр действителен только при активированном аппаратном управлении потоком. Действительный диапазон находится между 0 и 65535 мс. Значение 0 выключает функцию. (значение по умолчанию: 0)
BREAK	IN	UInt	Данный параметр указывает, что при запуске каждого сообщения на указанное количество битовых интервалов передается пауза. Максимум - это 65535 битовых интервалов до 8 секунд. Значение 0 выключает функцию. (значение по умолчанию: 12)
IDLELINE	IN	UInt	Данный параметр указывает, что перед запуском каждого сообщения на указанное количество битовых интервалов линия остается свободной. Максимум - это 65535 битовых интервалов до 8 секунд. Значение 0 выключает функцию. (значение по умолчанию: 0)
USR_END*	IN	STRING[2]	Определяет число и тип символов в конечном ограничителе. Конечный ограничитель встроен в буфер передачи (только символы) и обозначает конец переданного сообщения (символы передаются, пока не распознается конечный ограничитель). Конечный ограничитель добавляется в конце сообщения. • STRING[2,0,xx,yy] – ограничитель не используется (по умолчанию) • STRING[2,1,xx,yy] – ограничитель является одиночным символом • STRING[2,2,xx,yy] – ограничитель состоит из двух символов USR_END или APP_END должны иметь нулевую длину.

Параметр и тип данных		Тип данных	Описание
APP_END*	IN	STRING[5]	Определяет число и тип символов, которые будут добавлены к переданному сообщению (добавляются только символы). STRING[5,0,aa,bb,cc,dd,ee] – конечный символ не используется (по умолчанию) • STRING[5,1,aa,bb,cc,dd,ee] – передать один конечный символ • STRING[5,2,aa,bb,cc,dd,ee] – передать два конечных символов • STRING[5,3,aa,bb,cc,dd,ee]– передать три конечных символов • STRING[5,4,aa,bb,cc,dd,ee] – передать четыре конечных символов • STRING[5,5,aa,bb,cc,dd,ee] – передать пять конечных символов
DONE	OUT	Bool	Для целого цикла выполнения TRUE, после того как последний запрос был выполнен без ошибок.
ERROR	OUT	Bool	Для целого цикла выполнения TRUE, после того как последний запрос был выполнен с ошибками.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

* Не поддерживается для CM и CB 1241; следует использовать пустую строку ("") для параметра.

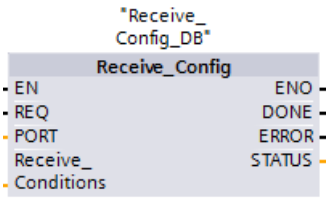
Таблица 13-12 Коды условий

STATUS (W#16#....)	Описание
81B0	Конфигурация прерывания передачи недопустима. Обратиться в службу технической поддержки.
81B1	Время паузы превышает максимально допустимое значение.
81B2	Время простоя превышает максимально допустимое значение.
81B3	Ошибка в заголовке блока (например, неправильный тип блока или неправильная длина блока)
81B4	Реконфигурирование отклонено, так как выполняется конфигурирование
81B5	Число определенных конечных ограничителей больше, чем два, или число конечных символов больше, чем пять
81B6	Конфигурации передачи отклоняется, если затрагивает встроенные в прошивку протоколы
8280	Отрицательное подтверждение при чтении из модуля
8281	Отрицательное подтверждение при записи в модуль
8282	Ведомое устройство DP или модуль недоступны

13.3.5.4 Инструкция Receive_Config (динамическое конфигурирование параметров для последовательного приема)

Таблица 13-13 Инструкция Receive_Config (конфигурация приема)

LAD/FBD	SCL	Описание

	`"Receive_Config_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_uint_in_,`
	
	`Receive_Conditions:=_struct_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		Инструкция Receive_Config выполняет динамическое конфигурирование параметров последовательного приема для PtP коммуникационного порта. Данная инструкция выполняет конфигурацию условий, которые обозначают начало и конец полученного сообщения. Любые сообщения в очереди в CM или CB сбрасываются, когда выполняется Receive_Config.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Можно настроить начальную статическую конфигурацию коммуникационного соединения в свойствах конфигурации устройства или просто использовать значения по умолчанию. После с помощью инструкции Receive_Config можно изменять конфигурацию в программе.

CPU не хранит постоянно значения, которые определяются с помощью инструкции Receive_Config. Параметры, сохраненные в конфигурации устройства, восстанавливаются CPU при переходе из RUN в STOP или при повторном включении. Дополнительная информация см. "Конфигурирование параметров приема (Страница 1128)".

Таблица 13-14 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Активирует изменение конфигурации при положительном фронте данного входа. (значение по умолчанию: False)
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC. (значение по умолчанию: 0)
CONDITIONS	IN	CONDITIONS	Структура данных Conditions определяет условия начала и окончания сообщения, как описано ниже.
DONE	OUT	Bool	TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок.
ERROR	OUT	Bool	TRUE в течение одного цикла, после того как последний запрос был выполнен с ошибками.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

Условия начала для инструкции Receive_P2P

Инструкция Receive_P2P использует конфигурацию, указанную в инструкции Receive_Config, чтобы определить начало и конец коммуникационных телеграмм "точка-точка". Начало сообщения определяется условиями начала. Начало сообщения определяется на основании одного или нескольких условий начала. Если указаны несколько условий начала, для запуска сообщения все условия должны быть соблюдены.

Описание условий для начала сообщения можно найти в "Конфигурирование параметров приема (Страница 1128)".

Структура типа данных параметра CONDITIONS, часть 1 (условия начала)

Таблица 13-15 Структура CONDITIONS для условий начала

Параметр и тип данных		Тип данных	Описание
STARTCOND	IN	UInt	Указание условий начала (значение по умолчанию: 1) • 01H - Стартовый символ • 02H - Любой символ • 04H - Пауза на линии • 08H - Свободная линия • 10H - Последовательность символов 1 • 20H - Последовательность символов 2 • 40H - Последовательность символов 3 • 80H - Последовательность символов 4
IDLETIME	IN	UInt	Число битовых интервалов, требуемых для тайм-аута свободной линии. (значение по умолчанию: 40). Только в сочетании с условием "Свободная линия". От 0 до 65535
STARTCHAR	IN	Byte	Начальный символ для условия "Начальный символ". (значение по умолчанию: B#16#2)
STRSEQ1CTL	IN	Byte	Последовательность символов 1, игнорировать/сравнить управление для каждого символа: (значение по умолчанию: B#16#0) Это биты разрешения для каждого символа в последовательности начала • 01H - символ 1 • 02H - символ 2 • 04H - символ 3 • 08H - символ 4 • 10H - символ 5 Деактивация бита, связанного с символом, означает, что любой символ будет подходящим в этой позиции последовательности.
STRSEQ1	IN	Char[5]	Последовательность символов 1, стартовые символы (5 символов). Значение по умолчанию: 0
STRSEQ2CTL	IN	Byte	Последовательность символов 2, игнорировать/сравнить управление для каждого символа: (значение по умолчанию: B#16#0)

Параметр и тип данных		Тип данных	Описание
STRSEQ2	IN	Char[5]	Последовательность символов 2, стартовые символы (5 символов). Значение по умолчанию: 0
STRSEQ3CTL	IN	Byte	Последовательность символов 3, игнорировать/сравнить управление для каждого символа: Значение по умолчанию: B#16#0
STRSEQ3	IN	Char[5]	Последовательность символов 3, стартовые символы (5 символов). Значение по умолчанию: 0
STRSEQ4CTL	IN	Byte	Последовательность символов 4, игнорировать/сравнить управление для каждого символа: Значение по умолчанию: B#16#0
STRSEQ4	IN	Char[5]	Последовательность символов 4, стартовые символы (5 символов), значение по умолчанию: 0

Пример

Рассмотрим следующее принятое шестнадцатерично-кодированное сообщение: "**68** 10 aa **68** bb 10 aa 16". Сконфигурированные последовательности начала показаны в таблице ниже. Последовательности начала анализируются, когда первый 68H символ успешно получен. После получения четвертого символа (второй 68H) условие начала 1 считается выполненным. После выполнения условий начала выполняется анализ условий завершения.

Обработка последовательности начала может быть завершена в результате различных ошибок четности, синхронизации кадров или интервалов между символами. Эти ошибки приводят к тому, что сообщение не будет принято, так как условие начала не было выполнено.

Таблица 13-16 Условия начала

Условие начала	Первый символ	Первый символ +1	Первый символ +2	Первый символ +3	Первый символ +4
1	**68**H	xx	xx	**68**H	xx
2	10H	aaH	xx	xx	xx
3	dcH	aaH	xx	xx	xx
4	e5H	xx	xx	xx	xx

Условия завершения для инструкции Receive_P2P

Конец сообщения определяется указанными условиями завершения. Конец сообщения определяется первым появлением одного или нескольких сконфигурированных условий завершения. В разделе "Условия для начала сообщения" в пункте "Конфигурирование параметров приема (Страница 1128)" описываются условия окончания, которые могут быть сконфигурированы для инструкции Receive_Config.

Выполнить конфигурацию условий завершения можно либо в свойствах коммуникационного интерфейса в конфигурации устройства, либо при помощи инструкции Receive_Config. Каждый раз при переходе CPU из STOP в RUN параметры приема (условия начала и завершения) снова переключаются на настройки конфигурации устройства. При выполнении программой пользователя STEP 7 инструкции Receive_Config, параметры устанавливаются на условия из Receive_Config.

Структура типа данных параметра CONDITIONS, часть 2 (условия завершения)

Таблица 13-17 Структура CONDITIONS для условий завершения

Параметр	Тип параметра	Тип данных	Описание
ENDCOND	IN	UInt 0	Этот параметр задает условие для конца сообщения: • 01H - Время отклика • 02H - Время сообщения • 04H - Интервал между символами • 08H - Максимальная длина • 10H - N + LEN + M • 20H - Последовательность символов
MAXLEN	IN	UInt 1	Максимальная длина сообщения: Используется только в том случае, если выбрано условие завершения "Максимальная длина". от 1 до 1024 байт
N	IN	UInt 0	Байтовая позиция поля длины в сообщении. Используется только с условием завершения N + LEN + M. от 1 до 1022 байт
LENGTHSIZE	IN	UInt 0	Размер поля длины (1, 2 или 4 байта). Используется только с условием завершения N + LEN + M.
LENGTHM	IN	UInt 0	Указать количество символов после поля длины, которые не входят в значение поля длины. Используется только с условием завершения N + LEN + M. от 0 до 255 байт
RCVTIME	IN	UInt 200	Указать время ожидания для первого полученного символа. Инструкция приема заканчивается ошибкой, если в течение указанного времени не будет выполнен прием символа. Такое указание используется только при условии "Время отклика". (от 0 до 65 535 битовых интервалов, максимум 8 секунд) Этот параметр не является условием завершения, так как анализ прерывается, когда первый символ отклика получен. Это является условием завершения только в том смысле, что приводит к завершению работы инструкции приема при отсутствии какого-либо ответа, когда ответ ожидается. Необходимо определить отдельное условие завершения.
MSGTIME	IN	UInt 200	Указать время ожидания полного приема всего сообщения после получения первого символа. Данный параметр используется только в том случае, если выбрано условие "Тайм-аут сообщения". (от 0 до 65 535 мс)
CHARGAP	IN	UInt 12	Указать количество битовых интервалов между символами. Если количество битовых интервалов превысит указанное значение, условие завершения считается выполненным. Такое указание используется только при условии "Интервал между символами". (от 0 до 65 535 битовых интервалов, максимум 8 секунд)

Параметр	Тип параметра	Тип данных	Описание
ENDSEQ1CTL	IN	Byte B#16#0	Последовательность символов 1, игнорировать/сравнить управление для каждого символа: Здесь указаны биты активации для каждого символа последовательность завершения. Символ 1 - бит 0, символ 2 - бит 1, ..., символ 5 - бит 4. Деактивация бита, связанного с символом, означает, что любой символ будет подходящим в этой позиции последовательности.
ENDSEQ1	IN	Char[5] 0	Последовательность 1, начальный символ (5 символов)

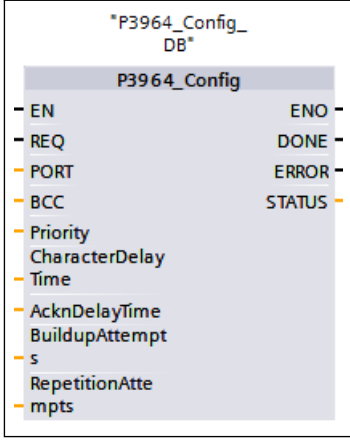
Таблица 13-18 Коды условий

STATUS (W#16#....)	Описание
81C0	Выбрано недопустимое условие начала
81C1	Недопустимое условие окончания или условие окончания не выбрано
81C2	Разрешено прерывание приема, и это невозможно.
81C3	Разрешено условие завершения по максимальной длине и максимальная длина равна 0 или > 1024.
81C4	Разрешена расчетная длина и N >= 1023
81C5	Разрешена расчетная длина и длина не равна 1, 2 или 4
81C6	Разрешена расчетная длина и значение M > 255.
81C7	Разрешена расчетная длина и она > 1024.
81C8	Разрешен тайм-аут отклика, и он равен нулю.
81C9	Разрешен тайм-аут межсимвольного интервала, и он равен нулю.
81CA	Разрешен тайм-аут свободной линии, и он равен нулю.
81CB	Последовательность завершения разрешена, но все символы, "игнорируются".
81CC	Последовательность начала (любая из 4) разрешена, но все символы, "игнорируются".
81CD	Ошибка защиты от перезаписи, недействительное принятое сообщение
81CE	Ошибка выбора обработки буфера неверно принятого сообщения при переходе из STOP в RUN
81CF	Ошибка в заголовке блока (например, неправильный тип блока или неправильная длина блока)
8281	Отрицательное подтверждение при записи в модуль
8282	Ведомое устройство DP или модуль недоступны
82C0	Реконфигурирование отклонено, так как выполняется конфигурирование
82C1	Заданное значение для числа сообщений, которые модуль может поместить в буфер, больше чем максимальное разрешенное значение.
82C2	Конфигурации приема отклоняется, если затрагивает встроенные в прошивку протоколы
8351	Тип данных недопустим для этого указателя Variant

13.3.5.5 P3964_Config (конфигурирование протокола 3964(R))

Таблица 13-19 Инструкция P3964_Config (3964_Config (конфигурирование протокола 3964(R))

LAD/FBD	SCL	Описание

	`"P3964_Config_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_uint_in_,`
	`BCC:=_usint_in,`
	`Priority:= _usint_in,`
	`CharacterDelayTime:=_uint_in,`
	`AcknDelayTime:= _uint_in,`
	`BuildupAttempts:=_usint_in_,`
	`RepetitionAttempts:=_usint_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		С помощью инструкции P3964_Config можно редактировать приоритет и параметры протокола при работе.
		Можно настроить начальную статическую конфигурацию порта в свойствах конфигурации устройства или просто использовать значения по умолчанию. С помощью инструкции P3964_Config можно изменять конфигурацию в программе.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

CPU не хранит постоянно значения, которые определяются с помощью инструкции P3964_Config. Параметры, сохраненные в конфигурации устройства, восстанавливаются CPU при повторном включении. Дополнительную информацию можно найти в Конфигурирование приоритета коммуникации и параметров протокола для 3964(R) (Страница 1137).

Таблица 13-20 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Активирует изменение конфигурации при положительном фронте данного входа. (значение по умолчанию: False)
PORT	IN	UInt	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC. (значение по умолчанию: 0)
BCC	IN	USInt	Активирует/деактивирует использование контроля блоков • 0 = без контроля блоков • 1 = с контролем блоков
Priority	IN	UInt	Выбор приоритета • 0 = низкий приоритет • 1 = высокий приоритет CM и партнер должны иметь противоположные приоритеты.
CharacterDelayTime	IN	UInt	Настройка времени задержки символа (зависит от настроенной скорости передачи данных) (значение по умолчанию: 220 мс) От 1 мс до 65 535 мс

Параметр и тип данных		Тип данных	Описание
AcknDelayTime	IN	UInt	Настройка времени задержки квитирования (зависит от настроенной скорости передачи данных) (значение по умолчанию: 2000 мс) От 1 мс до 65 535 мс
BuildupAttempts	IN	UInt	Количество попыток создания соединения (по умолчанию: 6 попыток соединения) 1 до 255
RepetitionAttempts	IN	UInt	Количество попыток передачи данных (по умолчанию: 6 попыток соединения) 1 до 255
DONE	OUT	Bool	Для целого цикла выполнения TRUE, после того как последний запрос был выполнен без ошибок.
ERROR	OUT	Bool	Для целого цикла выполнения TRUE, после того как последний запрос был выполнен с ошибками.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

Таблица 13-21 Коды условий

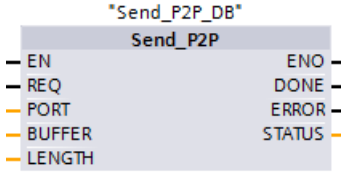
STATUS (W#16#....)	Описание
16#8380	Ошибка параметрирования: Недопустимое значение для "Времени задержки символа".
16#8381	Ошибка параметрирования: Недопустимое значение для "Тайм-аута ответа".
16#8382	Ошибка параметрирования: Недопустимое значение для "Приоритета".
16#8383	Ошибка параметрирования: Недопустимое значение для "Контроля блоков".
16#8384	Ошибка параметрирования: Недопустимое значение для "Попыток соединения".
16#8385	Ошибка параметрирования: Недопустимое значение для "Попыток передачи".
16#8386	Ошибка при выполнении программы: Количество попыток создания соединения превышено
16#8387	Ошибка при выполнении программы: Количество попыток передачи данных превышено
16#8388	Ошибка при выполнении программы: Ошибка в "Символе контроля блока" Вычисленное системой значение символа контроля блока не совпадает с полученным партнером на другом конце соединения символом контроля блока.
16#8389	Ошибка при выполнении программы: При ожидании свободного принимающего буфера получен недействительный символ.
16#838A	Ошибка при выполнении программы: Логическая ошибка при получении. После получения DLE был получен еще один случайный символ (не DLE или ETX).
16#838B	Ошибка при выполнении программы: Время задержки символа превышено
16#838C	Ошибка при выполнении программы: Началось время ожидания для свободного принимающего буфера
16#838D	Ошибка при выполнении программы: После NAK повтор телеграммы начинается не в течение 4 с
16#838E	Ошибка при выполнении программы: При простое был принят минимум один символ (не NAK или STX).

STATUS (W#16#....)	Описание
16#838F	Ошибка при выполнении программы: Конфликт инициализации - Оба партнера настроили высокий приоритет
16#8391	Ошибка параметрирования: Данные конфигурации 3964 отклонены, так как настроена свободно программируемая коммуникация

13.3.5.6 Инструкция Send_P2P (отправить данные из передающего буфера)

Таблица 13-22 Инструкция Send_P2P (передача данных точка-точка)

LAD/FBD	SCL	Описание

	`"Send_P2P_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_word_in_,`
	`BUFFER:=_variant_in_,`
	`LENGTH:=_uint_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		Инструкция Send_P2P запускает передачу данных и передает содержание назначенного буфера на коммуникационный интерфейс. Программа CPU продолжает работать, в то время как CM или CB отправляет данные с назначенной скоростью передачи. Одномоментно может быть активной только одна инструкция передачи. CM или CB возвращают ошибку, если вторая инструкция Send_P2P обрабатывается, в то время как CM или CB уже передают сообщение.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Таблица 13-23 Типы данных для параметров

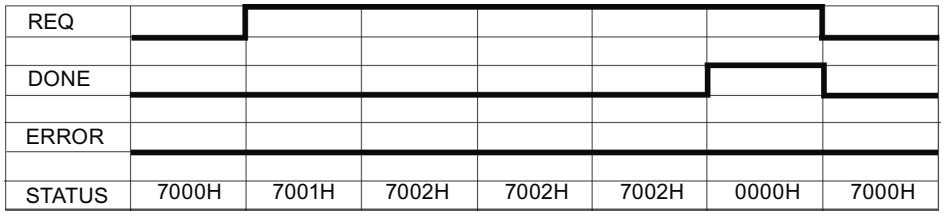
Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Разрешение на запрошенную передачу при положительном фронте этого разрешающего входа. Таким образом, содержание буфера передается на соответствующий коммуникационный интерфейс "точка-точка". (значение по умолчанию: False)
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC. (значение по умолчанию: 0)
BUFFER	IN	Variant	Этот параметр указывает на начальный адрес передающего буфера. (значение по умолчанию: 0) **Примечание:** Логические данные или логические поля не поддерживаются.
LENGTH	IN	UInt	Длина переданного кадра в байтах (значение по умолчанию: 0) При передаче сложной структуры всегда следует использовать длину 0. Когда длина 0, инструкция передает весь кадр.

Параметр и тип данных		Тип данных	Описание
DONE	OUT	Bool	TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок.
ERROR	OUT	Bool	TRUE в течение одного цикла, после того как последний запрос был выполнен с ошибками.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

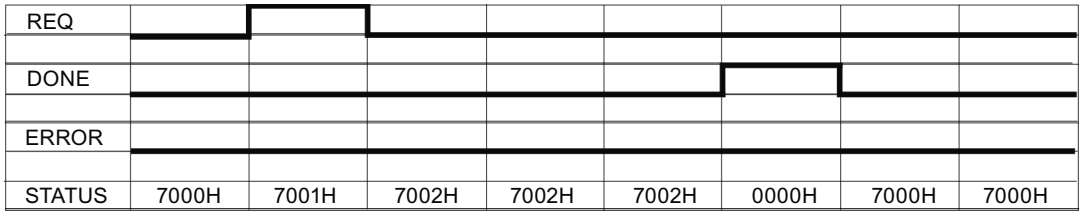
При обработке инструкции передачи, выходы DONE и ERROR находятся в состоянии FALSE. После окончания инструкции передачи один из выходов DONE или ERROR устанавливается на TRUE, чтобы сообщить о состояние инструкции передачи. Когда DONE или ERROR находятся в TRUE, выход STATUS допустим.

Инструкция возвращает состояние 16#7001, если коммуникационный интерфейс принимает переданные данные. Последующие выполнения Send_P2P возвращают 16#7002, если CM или CB все еще заняты передачей. Когда передача завершена, CM или CB возвращают состояние процесса передачи как 16#0000 (при отсутствии ошибок). Последующее выполнение Send_P2P с REQ = 0 возвращает состояние 16#7000 (не занято).

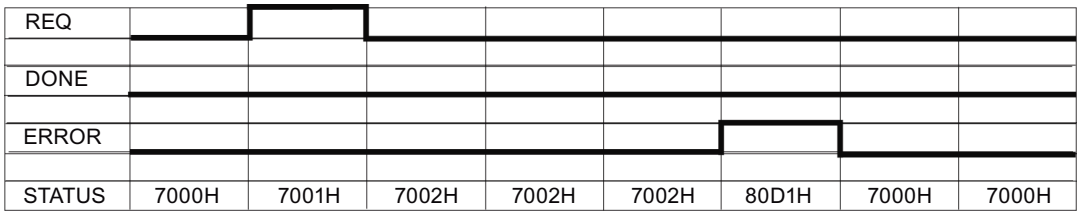
Следующая диаграмма показывает отношение выходных значений к REQ. Это предполагает, что инструкция вызывается периодически, чтобы проверить состояние процесса передачи. На диаграмме ниже предполагается, что инструкция вызывается в каждом цикле (представлено значениями STATUS).



Следующая диаграмма показывает, что параметры DONE и STATUS действительны только для одного цикла, если на линию REQ подается импульс (в течение одного цикла), чтобы инициировать процесс передачи.



Следующая диаграмма показывает отношение параметров DONE, ERROR и STATUS при ошибке.



Значения DONE, ERROR и STATUS допустимы, пока Send_P2P выполняется снова с тем же экземплярным DB.

Таблица 13-24 Коды условий

STATUS (W#16#....)	Описание
81D0	Новый запрос при активной передаче
81D1	Передача прервана из-за отсутствия CTS во время ожидания
81D2	Передача прервана из-за отсутствия DSR от DCE-устройства
81D3	Передача прервана из-за переполнения листа ожидания (передача более 1024 байтов)
81D5	Сигнал обратного смещения (обрыв провода)
81D6	Запрос передачи отклонен, так как конечный ограничитель не был найден в передающем буфере.
81D7	Внутренняя ошибка / ошибка синхронизации между FB и CM
81D8	Попытка передачи отклонена, так как порт не был сконфигурирован.
81DF	CM сбросил интерфейс к FB по одной из следующих причин: • Модуль перезапустился (цикл включения питания) • CPU достиг точки прерывания • Параметры модуля были заданы заново В каждом случае модуль указывает соответствующий код в параметре Status. Модуль сбрасывает Status и Error в после первой принятой записи для SEND_P2P.
8281	Отрицательное подтверждение при записи в модуль
8282	Ведомое устройство DP или модуль недоступны
8301	Недопустимый идентификатор синтаксиса в указателе ANY
8322	Ошибка длины области при считывании параметра
8324	Ошибка области при считывании параметра
8328	Ошибка выравнивания при считывании параметра
8332	Параметр содержит номер DB, который больше, чем наибольшее разрешенное число (ошибка номера DB).
833A	DB для параметра BUFFER не существует.

Примечание

Установка макс. длины записи для коммуникации Profibus

Если коммуникационный модуль CM1243-5 выступает в роли ведущего устройства Profibus для управления Profibus устройством ET200SP или ET200MP, которое в свою очередь используется модуль точка-точка RS232, RS422 или RS485, переменная (тег) блока данных "max_record_len" должна быть явно установлена на значение 240 следующим образом:

Установить после выполнения коммуникационной инструкции, напр., Port_Config, Send_Config или Receive_Config параметр max_record_len в DB экземпляра (напр., "Send_P2P_DB".max_record_len) на 240.

Явное присвоение значения для max_record_len необходимо только для Profibus соединений. Для Profinet соединений для max_record_len уже используется правильное значение.

Взаимодействие параметров LENGTH и BUFFER

Минимальным размером данных, которые могут быть отправлены инструкцией SEND_P2P, является один байт. Параметр BUFFER задает размер отправляемых данных. Для параметра BUFFER нельзя использовать ни тип данных Bool, ни массив типа Bool.

Можно в любое время установить параметр LENGTH на 0, гарантируя тем самым, что SEND_P2P передаст всю структуру данных параметра BUFFER. Если необходимо передать только часть структуры данных параметры BUFFER , можно установить LENGTH следующим образом:

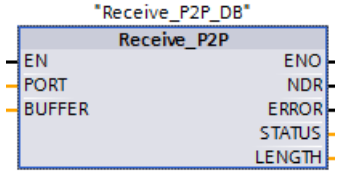
Таблица 13-25 Параметры LENGTH иBUFFER

LENGTH	BUFFER	Описание
= 0	Не используется	Полные данные отправляются согласно указаниям в параметре BUFFER. Если LENGTH = 0, количество передаваемых байтов указывать не надо.
>0	Элементарный тип данных	Значение LENGTH должно содержать количество байт этого типа данных. Пример: Для значения Word LENGTH должна составлять два. Для значения Dword или значения Real LENGTH должна составлять четыре. В противном случае передача выполняться не будет, и будет выведена ошибка 8088H.
	Структура	Значение LENGTH может иметь число байт, которое будет меньше полной длины байт структуры. В этом случае инструкцией передаются только первые n байт структуры из BUFFER,, при этом n = LENGTH. Так как внутренняя байтовая организация структуры не всегда может быть определена, это может привести к неожиданным результатам. Тогда следует использовать LENGTH = 0, чтобы передать структуру целиком.
	Массив	Значение LENGTHдолжно содержать число байт, которое меньше или равно полной байтовой длине массива, и которое должно быть кратным числу байт элемента данных. Пример: Параметр LENGTH массива типа Word должен быть кратным двум, а для массива типа Real - кратным четырем. Когда LENGTH определен, инструкция передает число элементов массива, которое соответствуют значению LENGTH в байтах. Если BUFFER, например, содержит массив из 15 элементов Dword (всего 60 байтов), и указывается LENGTH как 20, то передаются первые пять элементов Dword массива. Значение LENGTH должно быть кратным числу байт в элементе данных. В противном случае STATUS = 8088H, ERROR = 1 и передача не выполняется.
	String	Параметр LENGTH содержит количество отправляемых символов. Передаются только символы String. Байты с максимальной и фактической длиной String не передаются.

13.3.5.7 Инструкция Receive_P2P (разрешить прием сообщений)

Таблица 13-26 Инструкция Receive_P2P (прием данных точка-точка)

LAD/FBD	SCL	Описание

	`"Receive_P2P_DB"(`
	`PORT:=_word_in_,`
	`BUFFER:=_variant_in_,`
	`NDR=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`LENGTH=>_uint_out_);`
		Receive_P2P поверяет сообщения, принятые в CM или CB. Если сообщение доступно, то оно передается CM или CB на CPU. Ошибка возвращает соответствующее значение STATUS.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Таблица 13-27 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC. (значение по умолчанию: 0)
BUFFER	IN	Variant	Этот параметр указывает на начальный адрес принимающего буфера. Данный буфер должен иметь достаточно большой размер, чтобы принять телеграмму с максимальной длиной. Логические данные или логические поля не поддерживаются. (значение по умолчанию: 0)
NDR	OUT	Bool	Для целого цикла TRUE, если новые данные готовы и инструкция завершена без ошибок.
ERROR	OUT	Bool	Для целого цикла TRUE, после того как инструкция завершена с ошибкой.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)
LENGTH	OUT	UInt	Длина возвращенного сообщения в байтах (значение по умолчанию: 0)

Значение STATUS является действительным, если NDR или ERROR в состоянии TRUE. Значение STATUS описывает причину завершения операции приема в CM или CB. Обычно это положительное значение, указывающее на то, что операция приема была успешна и что процесс приема завершен обычным образом. Если значение STATUS отрицательное (устанавливается старший бит шестнадцатеричного значения), операция приема была завершена из-за ошибки, например ошибки четности, синхронизации кадров или переполнения.

Каждый PtP коммуникационный интерфейс может поместить в буфер максимум 1024 байта. Это может быть одним большим сообщением или несколькими маленькими сообщениями. Если больше чем одно сообщение доступно в CM или CB, инструкция

Receive_P2P возвращает самое старое доступное сообщение. Последующее выполнение инструкции Receive_P2P возвращает следующее самое старое доступное сообщение.

Таблица 13-28 Коды условий

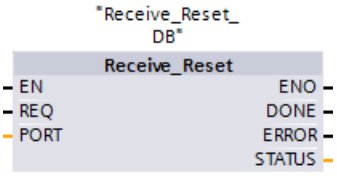
STATUS (W#16#...)	Описание
0000	Буфер не существует
0094	Сообщение было завершено, так как была принята максимальная длина строки
0095	Сообщение было завершено из-за тайм-аута сообщения
0096	Сообщение было завершено из-за межсимвольного тайм-аута
0097	Сообщение было завершено из-за тайм-аута ответа
0098	Сообщение было завершено, так как было выполнено условие длины "N+LEN+M".
0099	Сообщение было завершено, так как было выполнено условие длины "N+LEN+M".
8085	Значение параметра LENGTH составляет "0" или превышает 1 кБ.
8088	Параметр LENGTH или принятая длина больше, чем область, определенная в BUFFER, или принятая длина больше, чем область, определенная в BUFFER.
8090	Неверное сообщение о конфигурации, неправильная длина сообщения, неправильный субмодуль, недопустимое сообщение
81E0	Сообщение завершено, так как принимающий буфер заполнен
81E1	Сообщение было завершено из-за ошибки четности
81E2	Сообщение было завершено из-за ошибки кадровой синхронизации
81E3	Сообщение было завершено из-за ошибки переполнения
81E4	Сообщение было завершено, так как расчетная длина превышает размер буфера.
81E5	Сигнал обратного смещения (обрыв провода)
81E6	Лист ожидания с сообщениями заполнен. Эта ошибка сигнализируется без данных. Если это происходит, модуль переключается между безошибочной передачей данных и этой ошибкой.
81E7	Внутренняя ошибка, ошибка синхронизации между инструкцией и CM: устанавливается при обнаружении ошибки последовательности
81E8	Сообщение завершено, превышение времени между символами, прежде чем был определен конец сообщения
81E9	Обнаружена ошибка Modbus CRC (только для модулей, которые поддерживают генерацию/проверку CRC для протокола Modbus)
81EA	Телеграмма Modbus слишком короткая (только для модулей, которые поддерживают генерацию/проверку CRC для протокола Modbus)
81EB	Сообщение прервано, так как превышен максимальный размер сообщения
8201	Недопустимый идентификатор синтаксиса в указателе ANY
8223	Ошибка длины диапазона при записи параметра. Параметр расположен либо полностью, либо частично, вне диапазона адреса или длина битового диапазона не кратна 8 для указателя ANY.
8225	Ошибка диапазона при записи параметра. Параметр находится в диапазоне, недопустимом для системной функции.
8229	Ошибка выравнивания при записи параметра. Адресованный параметр расположен по битовому адресу, который не равен 0.
8230	Параметр находится в защищенном от записи глобальном блоке данных
8231	Параметр находится в защищенном от записи блоке данных экземпляра
8232	Параметр содержит номер DB, который больше, чем наибольший разрешенный номер блока (ошибка номера DB).

STATUS (W#16#...)	Описание
823A	DB для параметра BUFFER не существует.
8280	Отрицательное подтверждение при чтении из модуля
8282	Ведомое устройство DP или модуль недоступны

13.3.5.8 Инструкция Receive_Reset (удалить данные в принимающем буфере)

Таблица 13-29 Инструкция Receive_Reset (сброс принимающего устройства)

LAD/FBD	SCL	Описание

	`"Receive_Reset_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_word_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		Инструкция Receive_Reset очищает принимающий буфер в CM или CB.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

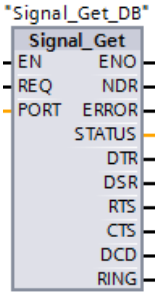
Таблица 13-30 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Активирует удаление данных в принимающем буфере при положительном фронте разрешающего входа (значение по умолчанию: False)
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC. (значение по умолчанию: 0)
DONE	OUT	Bool	Для целого цикла TRUE обозначает, что последний запрос был выполнен без ошибок.
ERROR	OUT	Bool	TRUE обозначает, что последний запрос был выполнен с ошибками. Если этот выход TRUE, выход STATUS содержит соответствующие коды ошибок.
STATUS	OUT	Word	Код ошибки (значение по умолчанию: 0)

13.3.5.9 Инструкция Signal_Get (запрос сигналов RS-232)

Таблица 13-31 Инструкция Signal_Get (получить сигналы RS232)

LAD/FBD	SCL	Описание

	`"Signal_Get_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_uint_in_,`
	`NDR=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`DTR=>_bool_out_,`
	`DSR=>_bool_out_,`
	`RTS=>_bool_out_,`
	`CTS=>_bool_out_,`
	`DCD=>_bool_out_,`
	`RING=>_bool_out_);`
		Signal_Get читает текущие состояния RS232 коммуникационных сигналов.
		Данная функция действительна только для RS232-CM.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Таблица 13-32 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Опрос состояния сигналов RS232 выполняется при положительном фронте на данном входе (значение по умолчанию: False)
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC.
NDR	OUT	Bool	Для целого цикла TRUE, если новые данные готовы и инструкция завершена без ошибок
ERROR	OUT	Bool	TRUE для целого цикла, после того как инструкция завершена с ошибкой.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)
DTR	OUT	Bool	Терминал данных готов, модуль готов (выход) Значение по умолчанию: False
DSR	OUT	Bool	Набор данных готов, участник обмена данными готов (вход). Значение по умолчанию: False
RTS	OUT	Bool	Запрос передачи, модуль готов к передаче (выход) Значение по умолчанию: False
CTS	OUT	Bool	Готов к передаче, участник обмена данными может принимать данные (вход). Значение по умолчанию: False
DCD	OUT	Bool	Носитель данных распознан, уровень сигнала получен (всегда False, не поддерживается)
RING	OUT	Bool	Индикатор вызова, сообщение входящего вызова (всегда False, не поддерживается)

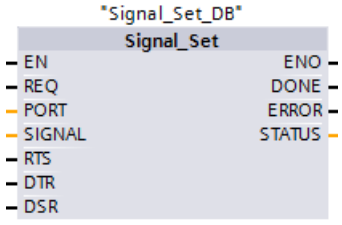
Таблица 13-33 Коды условий

STATUS (W#16#....)	Описание
81F0	CM или CB - это RS485 и нет доступных сигналов
81F4	Ошибка в заголовке блока (например, неправильный тип блока или неправильная длина блока)
8280	Отрицательное подтверждение при чтении из модуля
8282	Ведомое устройство DP или модуль недоступны

13.3.5.10 Инструкция Signal_Set (установка сигналов RS-232)

Таблица 13-34 Инструкция Signal_Set (установка сигналов RS232)

LAD/FBD	SCL	Описание

	`"Signal_Set_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_word_in_,`
	`SIGNAL:=_byte_in_,`
	`RTS:=_bool_in_,`
	`DTR:=_bool_in_,`
	`DSR:=_bool_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		Инструкция Signal_Set устанавливает состояния коммуникационных сигналов RS232.
		Данная функция действительна только для RS232-CM.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Таблица 13-35 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Инструкция для установки сигналов RS232 запускается при положительном фронте этого входа (значение по умолчанию: False)
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC. (значение по умолчанию: 0)
SIGNAL	IN	Byte	Выбирает сигналы для установки: (возможно несколько). Значение по умолчанию: 0 • 01H = RTS • 02H = DTR • 04H = DSR
RTS	IN	Bool	Запрос на передачу, модуль готов отправить значение для установки (true или false), значение по умолчанию: FALSE
DTR	IN	Bool	Терминал данных готов, модуль готов отправить значение для установки (true или false). Значение по умолчанию: FALSE

Параметр и тип данных		Тип данных	Описание
DSR	IN	Bool	Набор данных готов (действительно только для типа интерфейса DCE), не используется.
DONE	OUT	Bool	Для целого цикла выполнения TRUE, после того как последний запрос был выполнен без ошибок.
ERROR	OUT	Bool	Для целого цикла выполнения TRUE, после того как последний запрос был выполнен с ошибками.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

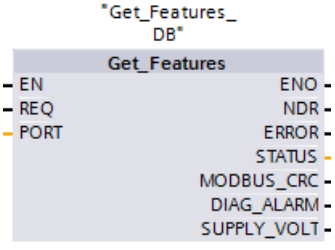
Таблица 13-36 Коды условий

STATUS (W#16#....)	Описание
81F0	CM или CB - это RS485 и никакие сигналы не могут быть установлены
81F1	Сигналы не могут быть установлены из-за аппаратного управления потоком
81F2	Нельзя установить DSR, так как модуль - DTE-устройство
81F3	Нельзя установить DTR, так как модуль - DTE-устройство
81F4	Ошибка в заголовке блока (например, неправильный тип блока или неправильная длина блока)
8280	Отрицательное подтверждение при чтении из модуля
8281	Отрицательное подтверждение при записи в модуль
8282	Ведомое устройство DP или модуль недоступны

13.3.5.11 Get_Features

Таблица 13-37 Инструкция Get_Features (опрос расширенных функций)

LAD/FBD	SCL	Описание

	`"Get_Features_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_word_in_,`
	`NDR:=_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`MODBUS_CRC=>_bool_out_,`
	`DIAG_ALARM=>_bool_out_,`
	`SUPPLY_VOLT=>_bool_out);`
		Инструкция Get_Features считывает расширенные функциональные возможности модуля.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Таблица 13-38 Типы данных для параметров

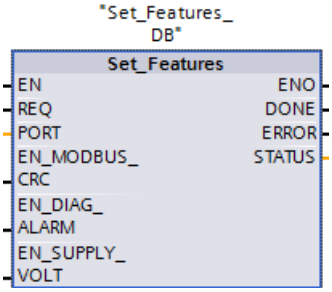
Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Активирует изменение в конфигурации при положительном фронте на данном входе. (значение по умолчанию: False)
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC. (значение по умолчанию: 0)
NDR	OUT	Bool	Показывает, что новые данные готовы.
ERROR	OUT	Bool	TRUE в течение одного цикла, после того как последний запрос был выполнен с ошибками.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)
MODBUS_CRC*	OUT	Bool	Генерация и проверка MODBUS CRC
DIAG_ALARM*	OUT	Bool	Создание диагностической тревоги
SUPPLY_VOLT*	OUT	Bool	Диагностика для отсутствующего напряжения питания L + доступна

*Get_Features возвращает TRUE (1), если функция доступна и FALSE (0), если функция недоступна

13.3.5.12 Set_Features

Таблица 13-39 Инструкция Set_Features (установка расширенных функций)

LAD/FBD	SCL	Описание

	`"Set_Features_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_word_in_,`
	`EN_MODBUS_CRC:=_bool_in_,`
	`EN_DIAG_ALARM:=_bool_in_,`
	`EN_SUPPLY_VOLT:=_bool_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		Set_Features определяет расширенные функции, которые поддерживает модуль.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Таблица 13-40 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Активирует изменение в конфигурации при положительном фронте на данном входе. (значение по умолчанию: False)
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC. (значение по умолчанию: 0)
EN_MODBUS_CRC	IN	Bool	Активация генерации и проверки MODBUS CRC: • 0: вычисление CRC отключено (по умолчанию) • 1: вычисление CRC включено Примечание: Только CM версии V2.1, CPU версии V4.1 с CB и CM модулями PtP для распределенного ввода-вывода поддерживают этот параметр.
EN_DIAG_ALARM	IN	Bool	Активировать диагностическую тревогу • 0: Диагностическая тревога ВЫКЛ • 1: Диагностическая тревога ВКЛ (по умолчанию)
EN_SUPPLY_VOLT	IN	Bool	Активировать диагностику для отсутствующего напряжения питания L +: • 0: Диагностика напряжение питания деактивирована (по умолчанию) • 1: Диагностика напряжение питания активирована
DONE	OUT	Bool	Указывает, что Set_Features выполнена
ERROR	OUT	Bool	TRUE в течение одного цикла, после того как последний запрос был выполнен с ошибками.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

13.3.6 Программирование PtP-коммуникации

STEP 7 предлагает расширенные инструкции, при помощи которых программа пользователя может выполнять коммуникацию "точка-точка" посредством протокола, разработанного и заданного в программе. Такие инструкции делятся на две категории:

- Инструкции конфигурирования
- Коммуникационные инструкции

Инструкции конфигурирования

Прежде чем программа пользователя сможет запустить коммуникацию PtP, необходимо выполнить конфигурирование коммуникационного интерфейса и параметров для передачи и приема данных.

Настройку интерфейса и сообщений для каждого CM или CB можно выполнить в конфигурации устройства или при помощи следующих инструкций программы пользователя:

- Port_Config (Страница 1141)
- Send_Config (Страница 1144)
- Receive_Config (Страница 1147)

Коммуникационные инструкции

При помощи инструкций для коммуникации "точка-точка" программа пользователя может отправлять сообщения на коммуникационные интерфейсы и получать сообщения от них. Дополнительную информация о передаче данных с помощью этих инструкций можно найти в разделе с информацией о целостности данных (Страница 191).

Все инструкции PtP работают асинхронно. При помощи архитектуры опроса программа пользователя может установить состояние передачи и приема. Send_P2P и Receive_P2P могут выполняться одновременно. В случае необходимости коммуникационные модули сохраняют в буфере переданные и полученные сообщения до достижения буфером максимального размера в 1024 байт.

CM и CB отправляют и принимают сообщения от участников обмена данными. Протокол сообщения находится в буфере, который определенный коммуникационный интерфейс отправляет или получает. Буфер и порт являются параметрами инструкций передачи и приема:

- Send_P2P (Страница 1154)
- Receive_P2P (Страница 1158)

При помощи дополнительных инструкций можно выполнить сброс принимающего буфера и опросить или установить специальные сигналы RS232:

- Receive_Reset (Страница 1160)
- Signal_Get (Страница 1161)
- Signal_Set (Страница 1162)

13.3.6.1 Архитектура опроса

Программа пользователя STEP 7 должна вызывать инструкции точка-точка S7-1200 циклически/периодически, чтобы выполнить проверку на принятые сообщения. Опрос отправления извещает программу пользователя о завершении передачи.

Архитектура опроса Ведущее устройство

Типичная последовательность для ведущего устройства:

1. Инструкция Send_P2P (Страница 1154) инициирует передачу на CM или CB.
2. Инструкция Send_P2P выполняется циклами, следующими друг за другом, чтобы выполнить опрос состояния процесса передачи.
3. Если инструкция Send_P2P сообщает, что передача данных завершена, код пользователя может подготовить получение ответа.

4. Инструкция Receive-P2P (Страница 1158) выполняется повторно, чтобы опросить ответ. После того как CM или CB получили ответное сообщение, инструкция Receive_P2P копирует ответ в CPU и сообщает, что были получены новые данные.
5. Программа пользователя может обработать ответ.
6. Затем выполняется возврат к шагу 1 и повтор цикла.

Архитектура опроса: Ведомое устройство

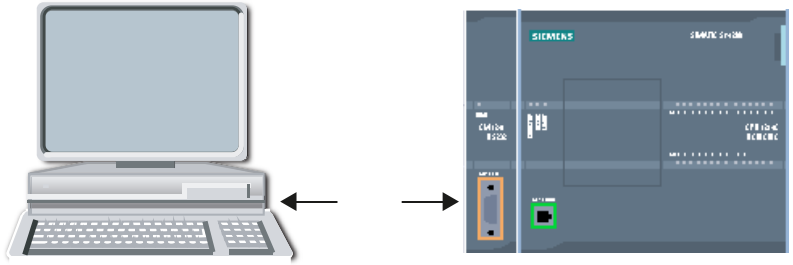
Типичная последовательность для ведомого устройства:

1. Программа пользователя выполняет инструкцию Receive_P2P в каждом цикле.
2. Если CM или CB получили запрос, инструкция Receive_P2P сообщает, что новые данные готовы и запрос копируется на CPU.
3. Программа пользователя обрабатывает запрос и создает ответ.
4. При помощи инструкции Send_P2P ответ отправляется обратно ведущему устройству.
5. Повторить Send_P2P , чтобы наверняка обеспечить процесс передачи.
6. Затем выполняется возврат к шагу 1 и повтор цикла.

Ведомое устройство должно обеспечить соответствующее количество вызовов Receive_P2P , чтобы получить передаваемые данные от ведущего устройства, прежде чем при ожидании ответа оно прервет процесс из-за превышения времени. Для выполнения этой задачи программа пользователя может вызывать RCV_PTP из OB цикла, время цикла которого достаточно длинное, чтобы получить передаваемые данные от ведущего устройства до истечения тайм-аута. Если время цикла для OB настроено таким образом, что в течение настроенного тайм-аута ведущего устройства возможно двукратное выполнение, программа пользователя может получить все передаваемые данные без потерь.

13.3.7 Пример: Коммуникации точка-точка

В этом примере S7-1200 CPU выполняет обмен данными с PC с эмулятором терминала через CM 1241 модуль RS232. Конфигурация точка-точка и программа STEP 7 в этом примере иллюстрируют, как CPU может получать сообщение от ПК и передавать его обратно на PC.



Необходимо подключить коммуникационный интерфейс CM 1241 модуля RS232 к интерфейсу RS232 компьютера. Обычно это COM1. Поскольку оба эти порта относятся к оконечному оборудованию данных (DTE), при соединении двух портов необходимо

переключить контакты приема и передачи (2 и 3). Это может быть выполнено одним из следующих способов:

- Использовать нуль-модемный адаптер, чтобы поменять местами контакты 2 и 3, вместе со стандартным кабелем RS232.
- Использовать нуль-модемный кабель, у которого уже перекинуты контакты 2 и 3. На концах нуль-модемного кабеля обычно располагаются два 9-штырьковых гнездовых D-соединителя.

13.3.7.1 Конфигурирование коммуникационного модуля

Можно сконфигурировать CM 1241 в конфигурации устройства в STEP 7 или с помощью инструкций в программе пользователя. Данный пример использует метод с конфигурацией устройства.

- Конфигурирование интерфейса: Кликнуть по коммуникационному порту модуля CM в конфигурации устройства и сконфигурировать порт как показано:

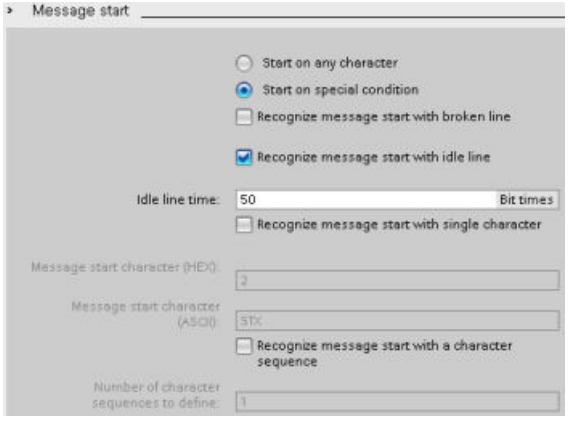


Примечание

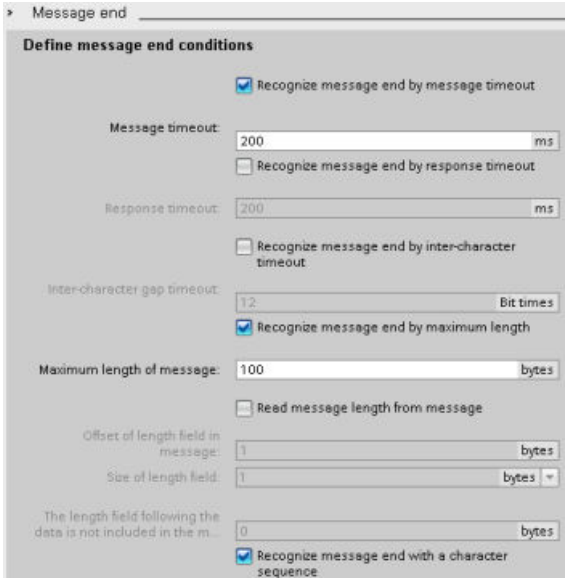
Настройки конфигурации для "Рабочего режима" и "Начального состояния линии приема" относятся только к модулю CM 1241 (RS422/RS485). У других модулей CM 1241 нет этих параметров конфигурации порта. Дополнительную информацию можно найти здесь: Конфигурирование RS422 и RS485 (Страница 1171).

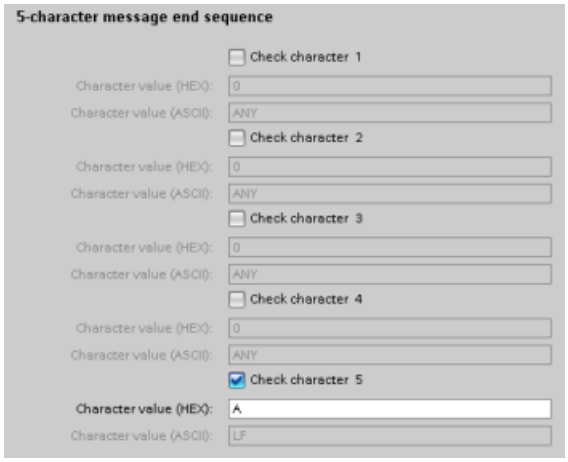
- Конфигурация передачи сообщения: Применить установку по умолчанию для конфигурации отправки сообщений. Пауза в начале сообщения не передается.

- Конфигурация начала приема сообщения: Сконфигурировать CM 1241 так, чтобы прием сообщения начинался тогда, когда линия связи будет неактивной в течение по крайней мере 50 битовых интервалов (приблизительно 5 миллисекунд при 9600 бодах = 50 * 1/9600):



- Конфигурация завершения приема сообщения: Сконфигурировать CM 1241 так, чтобы прием сообщения завершился после получения максимум 100 байтов или символа перевода строки (10 дес. или a шестн.). В качестве символьной последовательности завершения допускается до пяти конечных символов в последовательности. Пятый символ в последовательности является символом перевода строки. Предшествующие четыре конечных символа последовательности "игнорируются" или не выбираются. CM 1241 не анализирует "игнорируемые" символы, но ожидает символ перевода строки, которому предшествует 0, или больше "игнорируемых" символов, чтобы указать на завершение сообщения.




5-character message end sequence
Check character 1
Character value (HEX):
0
Character value (ASCII):
ANY
Check character 2
Character value (HEX):
0
Character value (ASCII):
ANY
Check character 3
Character value (HEX):
0
Character value (ASCII):
ANY
Check character 4
Character value (HEX):
0
Character value (ASCII):
ANY
Check character 5
Character value (HEX):
A
Character value (ASCII):
LF

13.3.7.2 Рабочие режимы RS422 и RS485

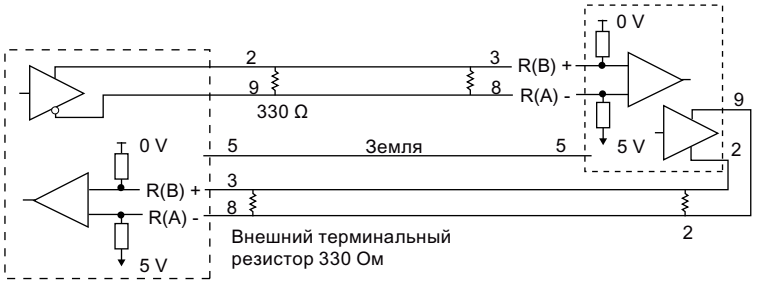
Конфигурирование RS422

Режим RS422, в зависимости от конфигурации сети, подразделяется на три рабочих режима. Следует выбрать один из этих рабочих режимов в зависимости от устройств в сети. Различные варианты выбора для начального состояние приемной линии относятся к представленным ниже случаям.

- Дуплексный режим (RS422), четырехпроводный режим (соединение точка-точка): Выбрать этот вариант, если сеть состоит из двух устройств. Для "Начального состояния линии приема":
  - Выбрать вариант "Без", если задаются смещение и завершение (случай 3).
  - Выбрать "Прямое смещение", чтобы использовать системные смещение и завершение (случай 2).
  - Выбрать "Обратное смещение", чтобы использовать системные смещение и завершение и включить обнаружение обрыва кабеля для обоих устройств (случай 1).
- Дуплексный режим (RS422), четырехпроводный режим (многоточечное ведущее устройство): Выбрать этот вариант, если сеть состоит из одного ведущего и нескольких ведомых устройств. Для "Начального состояния линии приема":
  - Выбрать вариант "Без", если задаются смещение и завершение (случай 3).
  - Выбрать "Прямое смещение", чтобы использовать системные смещение и завершение (случай 2).
  - Обнаружение обрыва кабеля в этом рабочем режиме невозможно.
- Дуплексный режим (RS422), четырехпроводный режим (многоточечное ведомое устройство): Выбрать этот вариант для всех ведомых устройств, если сеть состоит из одного ведущего и нескольких ведомых устройств. Для "Начального состояния линии приема":
  - Выбрать вариант "Без", если задаются смещение и завершение (случай 3).
  - Выбрать "Прямое смещение", чтобы использовать системные смещение и завершение (случай 2).
  - Выбрать "Обратное смещение", чтобы использовать системные смещение и завершение и включить обнаружение обрыва кабеля для ведомых устройств (случай 1).

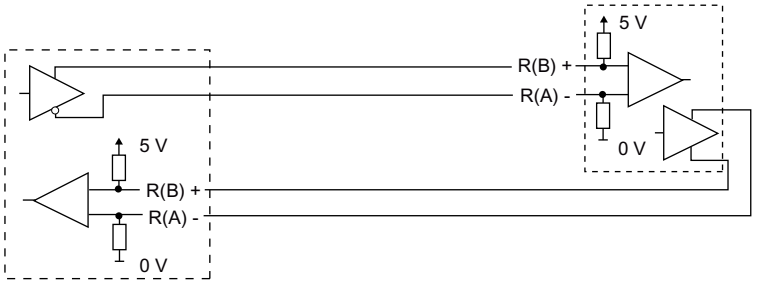
Случай 1: RS422 с обнаружением обрыва кабеля

- Режим работы: RS422
- Начальное состояние приемной линии: Обратное смещение (смещение с R(A) > R(B) > 0 В)
- Обрыв кабеля: Обнаружение обрыва кабеля активировано (передатчик всегда активен)



Случай 2: RS422 без обнаружения обрыва кабеля, прямое смещение

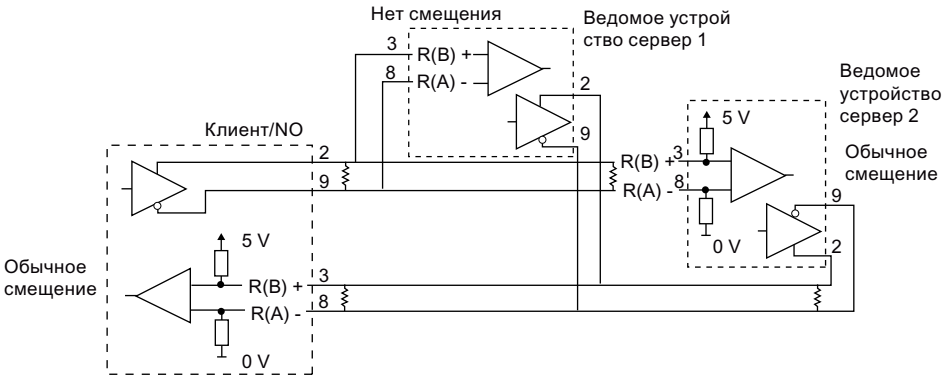
- Режим работы: RS422
- Начальное состояние приемной линии: Прямое смещение (смещение с R(B) > R(A) > 0 В)
- Обрыв кабеля: Без обнаружения обрыва кабеля (передатчик активируется только при передаче)



Случай 3: RS422: без обнаружения обрыва кабеля, без смещения

- Режим работы: RS422
- Начальное состояние приемной линии: Нет смещения
- Обрыв кабеля: Без обнаружения обрыва кабеля (передатчик активируется только при передаче)

Смещение и завершение добавляются пользователем в конечных узлах сети.



Конфигурирование RS485

Для режима RS485 есть только один рабочий режим. Различные варианты выбора для начального состояние приемной линии относятся к представленным ниже случаям.

- Полудуплексный режим (RS485), двухпроводный режим. Для "Начального состояния линии приема":
  - Выбрать вариант "Без", если задаются смещение и завершение (случай 5).
  - Выбрать "Прямое смещение", чтобы использовать системные смещение и завершение (случай 4).

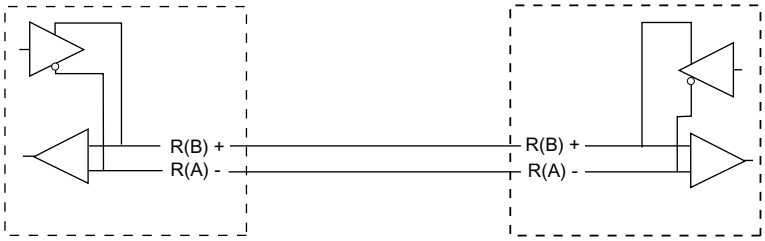
Случай 4: RS485: прямое смещение

- Режим работы: RS485
- Начальное состояние приемной линии: Прямое смещение (смещение с R(B) > R(A) > 0 В)



Случай 5: RS485: нет смещения (внешнее смещение)

- Режим работы: RS485
- Начальное состояние приемной линии: Нет смещения (требуется внешнее смещение)

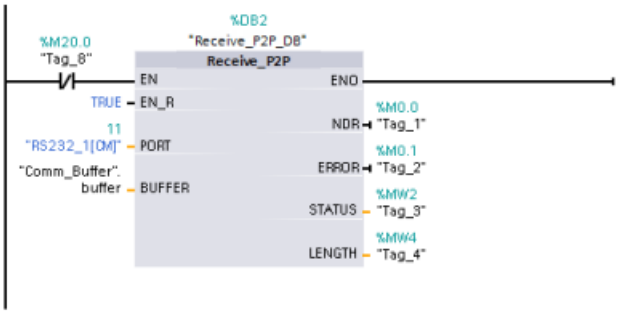


13.3.7.3 Создание программы STEP 7

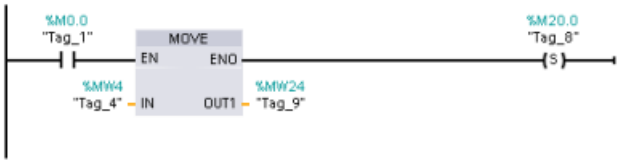
Программа-пример использует глобальный блок данных для коммуникационного буфера, инструкцию RCV_PTP (Страница 1309), чтобы получить данные от эмулятора терминала, и инструкцию SEND_PTP (Страница 1307), чтобы переслать эхо-буфер назад в эмулятор терминала. Чтобы запрограммировать пример, вставить конфигурацию блока данных и основной программный блок OB 1, как описано ниже.

Глобальный блок данных "Comm_Buffer": Создать глобальный блок данных (DB) и назвать его "Comm_Buffer". Создать один элемент в блоке данных с названием "buffer" с типом данных "Array [0 .. 99] of byte".

Сегмент 1: Активировать инструкцию RCV_PTP каждый раз, когда SEND_PTP не активна. Tag_8 в MW20.0 указывает в сегменте 4 на то, что передача завершена, и коммуникационный модуль готов принять сообщение.

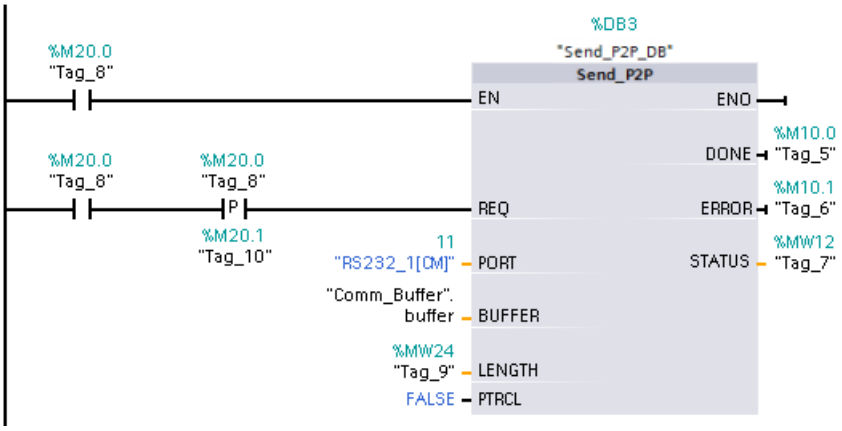


Сегмент 2: Создать с помощью установленного инструкцией RCV_PTP значенияNDR (Tag_1 в M0.0) копию количества принятых байт и установить флаг (Tag_8 в M20.0) для запуска инструкции SEND_PTP.

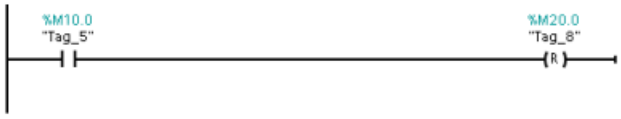


Сегмент 3: Активировать инструкцию SEND_PTP, если флаг M20.0 установлен. С помощью этого флага вход REQ устанавливается на TRUE на один цикл. Вход REQ сообщает инструкции SEND_PTP, что должен быть передан новый запрос. Вход REQ только в течение одного выполнения SEND_PTP может быть установлен на TRUE. Инструкция SEND_PTP выполняется в каждом цикле до завершения передачи. Передача завершена, когда последний байт сообщения был передан из CM 1241. Если передача завершена, то

выход DONE (Tag_5 в M10.0) в течение одного выполнения SEND_PTP устанавливается на TRUE.



Сегмент 4: Контролировать выход DONE в SEND_PTP и сбросить флаг передачи (Tag_8 в M20.0) после завершения процесса передачи. После сброса флага передачи, активируется инструкция RCV_PTP в сегменте 1, чтобы принять следующее сообщение.



13.3.7.4 Конфигурирование эмулятора терминала

Необходимо настроить эмулятор терминала для выполнения программы-примера. Можно использовать большую часть существующих эмуляторов терминала на своем PC, напр., HyperTerminal. Следует убедиться, что эмулятор терминала отключен прежде, чем редактировать настройки следующим образом:

1. Настроить эмулятор терминала для использования порта RS232 на PC (обычно COM1).
2. Сконфигурировать порт на 9600 бод, 8 битов данных, отсутствие контроля по четности, 1 стоповый бит и отсутствие управления потоком.
3. Изменить настройки эмулятора терминала, чтобы эмулировать терминал ANSI.
4. Сконфигурировать установку ASCII эмулятора терминала, чтобы посылать перевод строки после каждой строки (после того, как пользователь нажмет клавишу Enter).
5. Отображайте символы локально так, чтобы эмулятор терминала вывел на экран то, что введено.

13.3.7.5 Выполнение программы-примера

Для выполнения программы-примера действовать следующим образом:

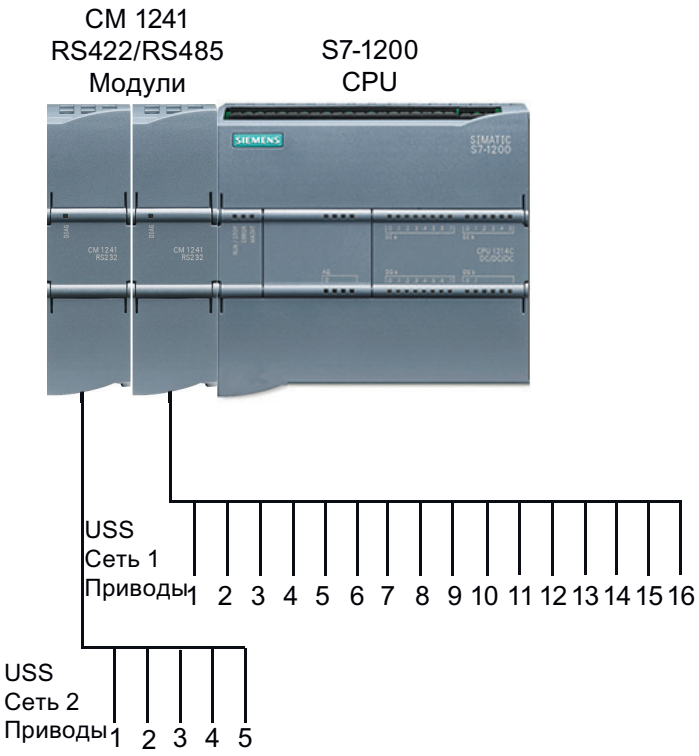
1. Загрузить STEP 7 программу в CPU и убедиться, что он находится в режиме RUN.
2. Нажать кнопку для соединения в эмуляторе терминала, чтобы применить изменения конфигурации и открыть сеанс работы на терминале для CM 1241.
3. Ввести символы на CPU и нажать кнопку ввода.

Эмулятор терминала передает символы на CM 1241 и на CPU. Программа CPU возвращает эхо символов на эмулятор терминала.

13.4 Коммуникация через универсальный последовательный интерфейс (USS)

Инструкции USS управляют работой электроприводов, которые поддерживают протокол универсального последовательного интерфейса (USS). Можно использовать USS-инструкции, чтобы связаться с несколькими приводами через RS485-соединения с коммуникационными модулями CM 1241 RS485 или коммуникационной платой CB 1241 RS485. До трех модулей CM 1241 RS422/RS485 и одна плата CB 1241 RS485 могут быть установлены в S7-1200 CPU. Каждый порт RS485 может работать макс. с 16 приводами.

Протокол USS использует сеть ведущее устройство - ведомое устройство для коммуникации по последовательной шине. Ведущее устройство использует один параметр адреса, чтобы отправлять сообщение выбранному ведомому устройству. Ведомое устройство никогда не может выполнять отправку, если оно предварительно не получило соответствующий запрос. Прямой обмен сообщениями между отдельными ведомыми устройствами невозможен. Коммуникация USS работает в полудуплексном режиме. На изображении ниже показан сетевой график для примера приложения с приводом.



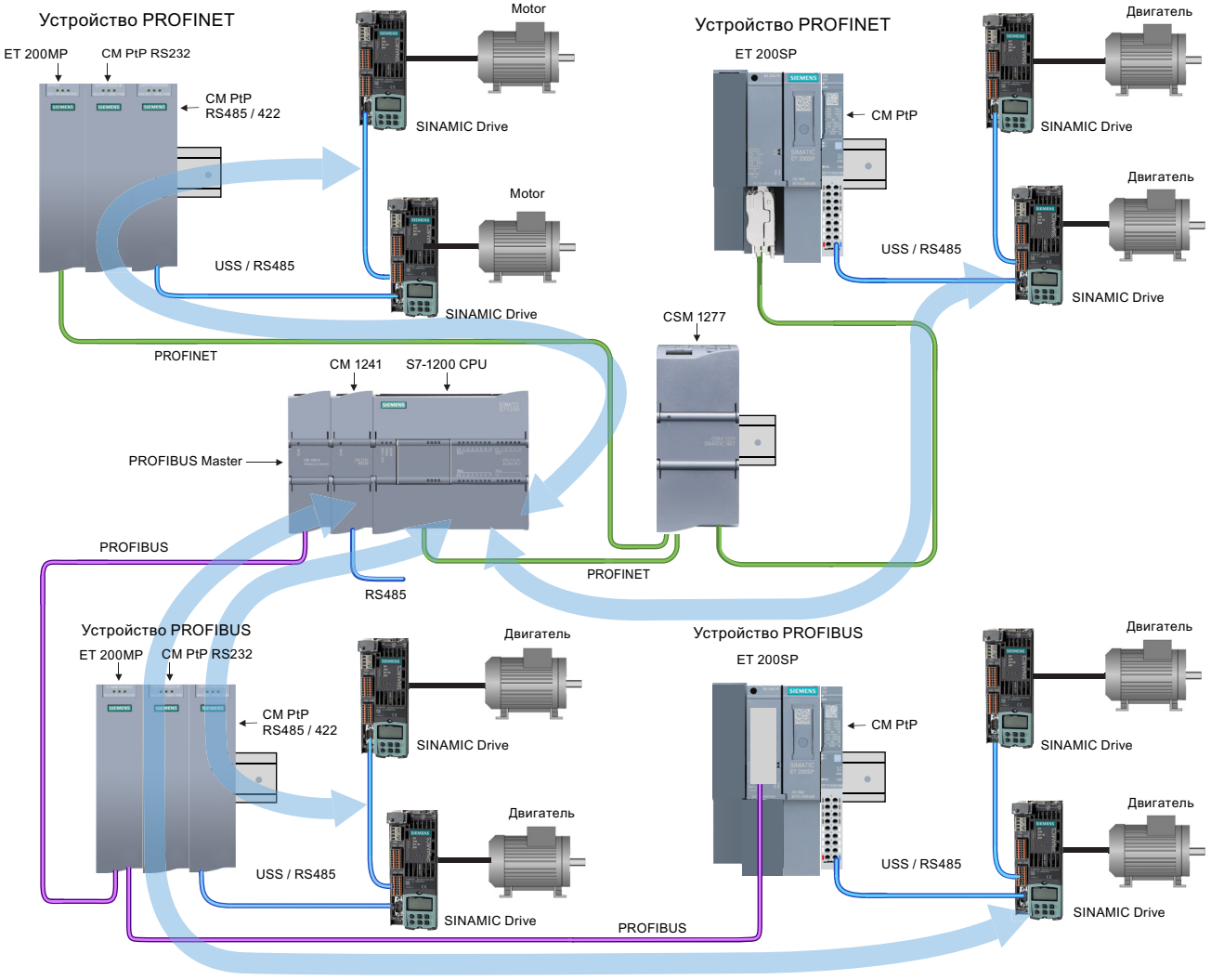
USS-коммуникации по PROFIBUS или PROFINET

От версии V4.1 S7-1200 CPU в комбинации с STEP 7 V13 SP1 расширяет возможности USS для CPU для использования стойки распределенного ввода-вывода PROFINET или PROFIBUS для связи с различными устройствами (RFID-считыватели, устройство GPS и другие):

- PROFINET (Страница 810): Интерфейс Ethernet S7-1200 CPU подключается к интерфейсному модулю PROFINET. Коммуникационные модули PtP в стойке с интерфейсным модулем могут в этом случае обеспечить последовательную коммуникацию с PtP-устройствами.
- PROFIBUS (Страница 984): Коммуникационный модуль PROFIBUS устанавливается на левой стороне стойки с S7-1200 CPU. Коммуникационный модуль PROFIBUS соединяется со стойкой, содержащей интерфейсный модуль PROFIBUS. Коммуникационные модули PtP в стойке с интерфейсным модулем могут в этом случае обеспечить последовательную коммуникацию с PtP-устройствами.

Поэтому S7-1200 поддерживает два набора PtP-инструкций:

- Старые инструкции USS (Страница 1318): Эти USS инструкции использовались до версии V4.0 S7-1200 и обеспечивают последовательную коммуникацию только с использованием коммуникационного модуля CM 1241 или коммуникационной платы CB 1241.
- Инструкции USS (Страница 1182): Эти USS инструкции обеспечивают всю функциональность старых инструкций плюс возможность подключения к распределенному вводу-выводу PROFINET и PROFIBUS. Эти USS инструкции позволяют конфигурировать связи между PtP коммуникационными модулями в стойке с распределенным вводом-выводом и PtP-устройствами. Для использования этих USS инструкций S7-1200 модулям CM 1241 потребуется как минимум прошивка версии V2.1.



Синяя стрелка обозначает двунаправленный поток данных между устройствами.

Примечание

От версии V4.1 S7-1200 инструкции PtP могут использоваться для любых типов коммуникации "точка-точка": последовательной, последовательной по PROFINET и последовательной по PROFIBUS. STEP 7 предлагает старые PtP инструкции только для поддержания имеющихся программ. Но старые инструкции работают на всех S7-1200 CPU. Преобразование старых программ под новые инструкции не требуется.

13.4.1 Выбор версии USS-инструкций

Существует две версии инструкций USS в STEP 7:

- Версия 2.0 (старые инструкции) изначально была доступна в STEP 7 Basic/Professional V13.
- Версия 2.1 доступна в STEP 7 Basic/Professional V13 SP1 и более поздних версий.

Учитывая совместимость и для упрощения миграции можно выбирать, какая версия инструкций будет вставлена в программу пользователя.

Нельзя использовать обе версии инструкций для одного модуля, но два различных модуля могут использовать различные версии инструкций.



Кликнуть по символу в окне задач с деревом инструкций, чтобы активировать заголовки и столбцы в дереве инструкций.



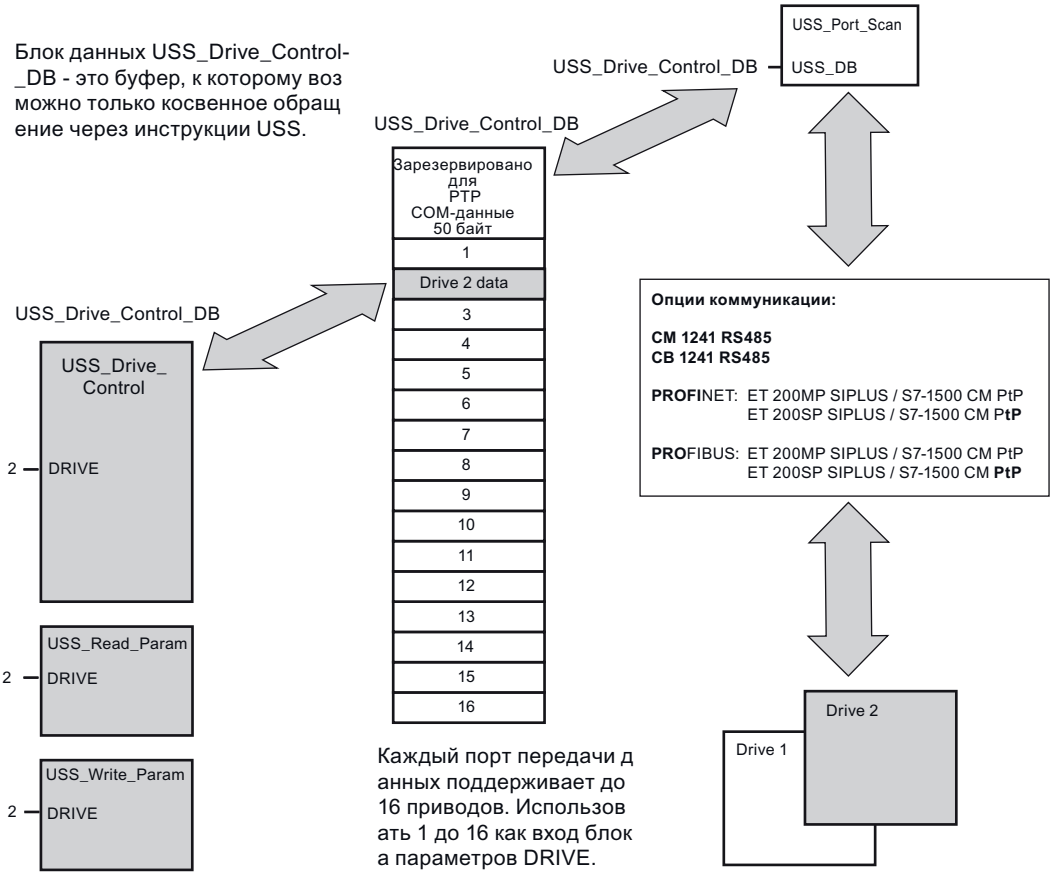
Для изменения версии USS инструкции, выбрать соответствующую версию в раскрывающемся списке. Можно выбрать группу или отдельные инструкции.

Если инструкция USS размещается через дерево инструкций в программе, то, в зависимости от выбранной инструкции USS, в дереве проекта создается новый экземпляр FB или FC. Для просмотра нового экземпляра FB или FC открыть в дереве проекта PLC_x > Программные блоки > Системные блоки > Программные ресурсы.

Чтобы узнать версию инструкции USS в программе, следует вызвать свойства в дереве проекта, а не свойства блочного элемента на экране в редакторе текстов программ. Выбрать в дереве проекта экземпляр FB или FC инструкции USS, кликнуть правой кнопкой мыши, выбрать "Свойства" и после страницу "Информация", чтобы отобразить номер версии инструкции USS.

13.4.2 Необходимые условия для использования протокола USS

Четыре USS-инструкции используют два функциональных блока (FB) и две функции (FC) для поддержки USS-протокола. Для каждой сети USS используется блок данных экземпляра USS_Port_Scan. Экземплярный блок данных USS_Port_Scan содержит временное хранилище и буферы для всех приводов в USS-сети. Инструкции USS имеют общий доступ к информации в этом блоке данных.



Все приводы (макс. 16), подключенные к одному порту RS485, являются частью одной сети USS. Все приводы, подключенные к другому порту RS485, являются частью другой сети USS. Каждая сеть USS управляется при помощи уникального блока данных. Все инструкции, относящиеся одной сети USS, должны совместно использовать этот блок данных. Он включает все инструкции USS_Drive_Control, USS_Port_Scan, USS_Read_Param и USS_Write_Param для управления всеми приводами в сети USS.

Инструкция USS_Drive_Control является функциональным блоком (FB). Если добавить инструкцию USS_Drive_Control в редактор текстов программ, в диалоговом окне "Опции вызова" придется для этого FB присвоить DB. Если речь идет о первой инструкции USS_Drive_Control в этой программе для этой сети USS, можно применить стандартную инструкцию блока данных (или при необходимости изменить имя), и новый блок данных будет готов. Если же это не первая инструкция USS_Drive_Control для этого канала, в диалоговом окне "Опции вызова" в выпадающем списке необходимо выбрать блок данных, который уже был назначен сети USS.

Инструкция USS_Port_Scan является функциональным блоком (FB) и обрабатывает фактический обмен между CPU и приводами через коммуникационный порт RS485 "точка-точка" (PtP). При каждом вызове этого FB обрабатывается коммуникация с одним приводом. В программе этот FB должен вызываться достаточно быстро для того, чтобы приводы не сообщали о превышении времени. Этот FB может быть вызван из OB цикла основной программы или из любого OB прерываний.

Обе инструкции USS_Read_Param и USS_Write_Param являются функциями (FC). Если вставлять эти функции в редакторе, DB не присваивается. Вместо этого входу USS_DB этих инструкций необходимо назначить соответствующий DB. Выполнить двойной клик по полю параметра и затем кликнуть по символу, чтобы отобразить доступные блоки данных.

Как правило, вызывается FB USS_Port_Scan из OB циклического прерывания. Время цикла для OB циклического прерывания должно быть установлено приблизительно на половину минимального интервала вызова (например, обмен данными на скорости 1200 бод должен использовать время цикла в 350 мс или меньше).

Функциональный блок USS_Drive_Control предоставляет программе доступ к указанному приводу в сети USS. Его входы и выходы соответствуют состояниям и функциям управления привода. Если в сети имеется 16 приводов, в программе необходимо вызвать USS_Drive_Control не менее 16 раз, т. е. один раз для каждого привода. Скорость вызова этих блоков зависит от необходимой скорости для управления функциями приводов.

FB USS_Drive_Control можно вызывать только из OB цикла главной программы.

⚠ ОСТОРОЖНО

Вызов инструкций USS из OB

Вызывать USS_Drive_Control, USS_Read_Param и USS_Write_Param только из OB цикла главной программы. FB USS_Port_Scan можно вызывать из любого OB, обычно он вызывается из OB циклического прерывания.

Не использовать инструкции USS_Drive_Control, USS_Read_Param и USS_Write_Param в OB с более высоким приоритетом, чем соответствующая инструкция USS_Port_Scan. Не следует вставлять, например USS_Port_Scan в OB главной программы, а USS_Read_Param в OB циклического прерывания. Отказ предотвратить прерывание обработки USS_Port_Scan может привести к неожиданным ошибкам, которые в свою очередь могут стать причиной телесных повреждений.

С помощью функций USS_Read_Param и USS_Write_Param считываются и записываются рабочие параметры удаленного привода. Эти параметры управляют внутренним принципом действия привода. Определение этих параметров см. в руководстве по эксплуатации привода. В программе может быть любое количество этих функций, но для одного привода всегда должен быть активен только один запрос считывания или записи. FC USS_Read_Param и USS_Write_Param можно вызывать только из OB цикла главной программы.

Расчет времени для коммуникации с приводом

Коммуникация с приводом выполняется асинхронно по отношению к циклу S71200. S7-1200 обычно проходит несколько циклов, прежде чем будет завершена коммуникация с приводом.

Промежуток USS_Port_Scan является временем, необходимым для одной транзакции привода. В следующей таблице показаны минимальные интервалы для USS_Port_Scan для каждой скорости передачи данных. Если вызывать FB USS_Port_Scan чаще, чем предписывает интервал USS_Port_Scan, количество транзакций не увеличивается. Интервал тайм-аута привода - это промежуток времени, который доступен для транзакции, если для ее завершения из-за ошибок коммуникации требуется 3 попытки. По умолчанию библиотека USS протокола автоматически делает до 2 повторений на каждой транзакции.

Таблица 13-41 Расчет необходимого времени

Скорость передачи	Рассчитанный минимальный интервал для вызова USS_Port_Scan (мс)	Тайм-аут интервала для сообщения на каждый привод (мс)
1200	790	2370
2400	405	1215
4800	212.5	638
9600	116.3	349
19200	68.2	205
38400	44.1	133
57600	36.1	109
115200	28.1	85

13.4.3 Инструкции USS

13.4.3.1 USS_Port_Scan (обработка коммуникации через сеть USS)

Таблица 13-42 Инструкция USS_Port_Scan

LAD/FBD	SCL	Описание

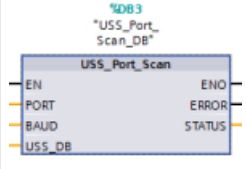
	`USS_Port_Scan(`
	`  PORT:=_uint_in_,`
	`  BAUD:=_dint_in_,`
	`  ERROR=>_bool_out_,`
	`  STATUS=>_word_out_,`
	`  USS_DB:=_fbtref_inout_);`
		Инструкция USS_Port_Scan обрабатывает коммуникацию по сети USS.

Таблица 13-43 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
PORT	IN	Port	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC.
BAUD	IN	DInt	Скорость передачи данных для коммуникации USS.
USS_DB	INOUT	USS_BASE	Имя блок данных экземпляра, который создается и инициализируется при вставке инструкции USS_Drive_Control в программу.
ERROR	OUT	Bool	Если TRUE, этот выход указывает на ошибку и действительным является выход STATUS.
STATUS	OUT	Word	Значение состояния для запроса отображает результат цикла или инициализации. Дополнительная информация для отдельных кодов состояния доступна в переменной "USS_Extended_Error".

Обычно в программе для каждого коммуникационного порта PtP существует по одной инструкции USS_Port_Scan , и каждый вызов этого функционального блока (FB) управляет передачей данных на один привод этой сети или из него. Все функции USS, назначенные сети USS и коммуникационному порту PtP, должны использовать одинаковый экземплярный блок данных.

Программа должна выполнять инструкцию USS_Port_Scan достаточно часто, чтобы предотвратить превышение времени в приводе. USS_Port_Scan обычно вызывают из OB циклического прерывания, чтобы предотвратить превышение времени приводов и сохранить последние обновления USS-данных доступными для вызовов USS_Drive_Control.

Примечание

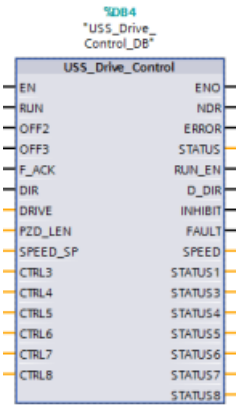
При использовании библиотеки для протокола USS и инструкции USS_Port_Scan с CB 1241, для переменной блока данных LINE_PRE должно быть установлено значение "0 (не исходное состояние). Значение по умолчанию 2 для переменной блока данных LINE_PRE приводит к возврату значения ошибки 16#81AB инструкцией USS_Port_Scan. Переменную блока данных LINE_PRE можно найти в блоке данных инструкции USS_Port_Scan (как правило, под именем USS_Port_Scan_DB).

Проследить, чтобы начальное значение LINE_PRE было изменено на 0 (ноль).

13.4.3.2 Инструкция USS_Drive_Control (обмен данными с приводом)

Таблица 13-44 Инструкция USS_Drive_Control

LAD/FBD	SCL	Описание

	`"USS_Drive_Control_DB"(`
	`RUN:=_bool_in_,`
	`OFF2:=_bool_in_,`
	`OFF3:=_bool_in_,`
	`F_ACK:=_bool_in_,`
	`DIR:=_bool_in_,`
	`DRIVE:=_usint_in_,`
	`PZD_LEN:=_usint_in_,`
	`SPEED_SP:=_real_in_,`
	`CTRL3:=_word_in_,`
	`CTRL4:=_word_in_,`
	`CTRL5:=_word_in_,`
	`CTRL6:=_word_in_,`
	`CTRL7:=_word_in_,`
	`CTRL8:=_word_in_,`
	`NDR=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`RUN_EN=>_bool_out_,`
	`D_DIR=>_bool_out_,`
	`INHIBIT=>_bool_out_,`
	`FAULT=>_bool_out_,`
	`SPEED=>_real_out_,`
	`STATUS1=>_word_out_,`
	`STATUS3=>_word_out_,`
	`STATUS4=>_word_out_,`
	`STATUS5=>_word_out_,`
	`STATUS6=>_word_out_,`
	`STATUS7=>_word_out_,`
	`STATUS8=>_word_out_);`
		Инструкция USS_Drive_Control обменивается данными с приводом, создавая сообщения запроса и анализируя ответные сообщения привода. Для каждого привода необходимо использовать свой функциональный блок, но все функции USS, назначенные сети USS и коммуникационному порту PtP, должны использовать один экземплярный блок данных. При добавлении первой инструкции USS_Drive_Control необходимо ввести имя блока данных. Затем указать ссылку на этот блок данных, который был создан при вставке первой инструкции.
		При вставке инструкции STEP 7 автоматически создает DB.

[1] LAD и FBD: Развернуть блок, чтобы просмотреть все параметры. Для этого кликнуть мышью по нижней области блока. Соединители параметров, отображенные серым цветом, являются опциональными, как и их параметрирование.

Таблица 13-45 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
RUN	IN	Bool	Стартовый бит привода: Если этот параметр TRUE, этот вход позволяет приводу работать с предустановленной скоростью. Если во время работы привода RUN меняется на false, двигатель выполняет выбег до остановки. Такое поведение отличается от отключения питания (OFF2) и от торможения двигателя (OFF3).
OFF2	IN	Bool	Бит "Выбег до останова": Если этот параметр FALSE, этот бит допускает инерционный выбег привода без торможения.
OFF3	IN	Bool	Быстрый стоповый бит: Если этот параметр FALSE, этот бит вызывает быстрый останов с помощью торможения привода.

Параметр и тип данных		Тип данных	Описание
F_ACK	IN	Bool	Бит квитирования ошибки: С помощью этого бита сбрасывается бит ошибки привода. Бит устанавливается после удаления ошибки, и таким образом привод распознает, что о предыдущей ошибке больше сообщать не надо.
DIR	IN	Bool	Управление направлением привода: Этот бит устанавливается, если привод должен вращаться вперед (при положит. SPEED_SP).
DRIVE	IN	USInt	Адрес привода: Этот вход является адресом привода USS. Действительный диапазон находится между приводом 1 и приводом 16.
PZD_LEN	IN	USInt	Длина в словах: Указывает на количество слов данных PZD. Действительными значениями являются 2, 4, 6 или 8 слов. Стандартное значение равно 2.
SPEED_SP	IN	Real	Заданное значение скорости: Это скорость привода как процент заданной частоты. Положительное значение говорит о том, что привод вращается вперед (при истинном DIR). Действительный диапазон составляет от 200,00 до -200,00.
CTRL3	IN	Word	Управляющее слово 3: Значение, которое записывается в конфигурируемый пользователем параметр привода. Конфигурация выполняется в приводе (опциональный параметр).
CTRL4	IN	Word	Управляющее слово 4: Значение, которое записывается в конфигурируемый пользователем параметр привода. Конфигурация выполняется в приводе (опциональный параметр).
CTRL5	IN	Word	Управляющее слово 5: Значение, которое записывается в конфигурируемый пользователем параметр привода. Конфигурация выполняется в приводе (опциональный параметр).
CTRL6	IN	Word	Управляющее слово 6: Значение, которое записывается в конфигурируемый пользователем параметр привода. Конфигурация выполняется в приводе (опциональный параметр).
CTRL7	IN	Word	Управляющее слово 7: Значение, которое записывается в конфигурируемый пользователем параметр привода. Конфигурация выполняется в приводе (опциональный параметр).
CTRL8	IN	Word	Управляющее слово 8: Значение, которое записывается в конфигурируемый пользователем параметр привода. Конфигурация выполняется в приводе (опциональный параметр).
NDR	OUT	Bool	Новые данные готовы: При истинном значении этого параметра бит сообщает, что на выходе готовы данные нового запроса коммуникации.
ERROR	OUT	Bool	Возникла ошибка: Если TRUE, это указывает на ошибку, и действительным является выход STATUS. Все остальные выходы при возникновении ошибки устанавливаются на ноль. Об ошибках коммуникации сообщается только на выходах ERROR и STATUS инструкции USS_Port_Scan.
STATUS	OUT	Word	Значение состояния запроса отображает результат цикла. Это не слово состояния, возвращаемое приводом.
RUN_EN	OUT	Bool	Работа разрешена: Этот бит сообщает о работе привода.
D_DIR	OUT	Bool	Направление привода: Этот бит сообщает, вращается ли привод вперед.
INHIBIT	OUT	Bool	Привод заблокирован: Этот бит сообщает о состоянии бита блокировки для привода.

Параметр и тип данных		Тип данных	Описание
FAULT	OUT	Bool	Ошибка привода: Этот бит сообщает, что в приводе возникла ошибка. Необходимо устранить неисправность и установить бит F_ACK, чтобы удалить этот бит.
SPEED	OUT	Real	Фактическое значение скорости привода (масштабированное значение от слова состояния 2 привода): значение скорости привода, как процент от заданной скорости.
STATUS1	OUT	Word	Слово состояния 1 привода: Данное значение включает в себя фиксированные биты состояния привода.
STATUS3	OUT	Word	Слово состояния 3 привода: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS4	OUT	Word	Слово состояния 4 привода: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS5	OUT	Word	Слово состояния 5 привода: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS6	OUT	Word	Слово состояния 6 привода: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS7	OUT	Word	Слово состояния 7 привода: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS8	OUT	Word	Слово состояния 8 привода: Это значение содержит конфигурируемое пользователем слово состояния привода.

При первом выполнении инструкции USS_Drive_Control, инициализируется указанный по адресу USS (параметр DRIVE) привод в экземплярном блоке данных. После инициализации последующие инструкции USS_Port_Scan могут начинать коммуникацию с приводом по этому номеру привода.

Если изменить номер привода, CPU сначала необходимо перевести в состояние STOP и затем обратно в RUN, чтобы инициализировать экземплярный блок данных. Входные параметры конфигурируются в передающем буфере USS, а выходы, если имеются, считываются из "предыдущего" действительного буфера ответа. Во время выполнения инструкции USS_Drive_Control передача данных не выполняется. После выполнения USS_Port_Scan приводы выполняют обмен данными. USS_Drive_Control конфигурирует только отправляемые сообщения и анализирует данные, которые возможно были получены в предыдущем запросе.

Направлением вращения привода можно управлять с помощью входа DIR (Bool) или с помощью знака (плюс или минус) на входе SPEED_SP (Real). В следующей таблице дается пояснение того, как взаимодействуют входы, чтобы задавать направление вращения привода, при условии, что двигатель вращается вперед.

Таблица 13-46 Взаимодействие параметров SPEED_SP и DIR

SPEED_SP	DIR	Направление вращения привода
Значение > 0	0	Назад
Значение > 0	1	Вперед
Значение < 0	0	Вперед
Значение < 0	1	Назад

13.4.3.3 Инструкция USS_Read_Param (считывание параметров из привода)

Таблица 13-47 Инструкция USS_Read_Param

LAD/FBD	SCL	Описание

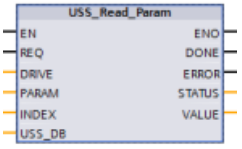
	`USS_Read_Param(REQ:=_bool_in_,`
	`  DRIVE:=_usint_in_,`
	`  PARAM:=_uint_in_,`
	`  INDEX:=_uint_in_,`
	`  DONE=>_bool_out_,`
	`  ERROR=>_bool_out_,`
	`  STATUS=>_word_out_,`
	`  VALUE=>_variant_out_,`
	`  USS_DB:=_fbtref_inout_);`
		Инструкция USS_Read_Param считывает параметр из привода. Все функции USS, назначенные сети USS и коммуникационному порту PtP, должны использовать один блок данных. USS_Read_Param должен вызываться из циклического OB главной программы.

Таблица 13-48 Типы данных для параметров

Тип параметра		Тип данных	Описание
REQ	IN	Bool	Запрос на передачу: Если REQ имеет значение TRUE, требуется новый запрос на чтение. Он игнорируется, если уже есть активный запрос для этого параметра.
DRIVE	IN	USInt	Адрес привода: DRIVE является адресом привода USS. Действительный диапазон находится между приводом 1 и приводом 16.
PARAM	IN	UInt	Номер параметра: PARAM указывает, какой параметр привода записывается. Диапазон для этого параметра составляет от 0 до 2047. В некоторых приводах старший значащий байт может получить доступ к значениям PARAM, больше, чем 2047. Дополнительная информация о доступе к расширенному диапазону находится в руководстве по эксплуатации привода.
INDEX	IN	UInt	Индекс параметра: INDEX указывает, в какой индекс параметра привода должна выполняться запись. Речь идет о 16-битном значении, в котором младший байт является фактическим значением индекса, с диапазоном (от 0 до 255). Старший байт также может использоваться приводом и зависит от типа привода. Дополнительная информация находится в руководстве по эксплуатации привода.
USS_DB	INOUT	USS_BASE	Имя блок данных экземпляра, который создается и инициализируется при вставке инструкции USS_Drive_Control в программу.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDInt, Real	Это значение параметра, которое было считано, и оно действительно только в том случае, если бит DONE имеет истинное значение.
DONE[1]	OUT	Bool	Если этот параметр TRUE, на выходе VALUE будет запрошенное ранее значение параметра для чтения. Этот бит устанавливается, если инструкция USS_Drive_Control распознает ответ считывания привода. Этот бит сбрасывается, если: запрашиваются ответные данные, используя другой запрос USS_Read_Param, либо при втором из следующих двух вызовов USS_Drive_Control.

Тип параметра		Тип данных	Описание
ERROR	OUT	Bool	Возникла ошибка: Если TRUE, ERROR указывает на ошибку и действительный выход STATUS. Все остальные выходы при возникновении ошибки устанавливаются на ноль. Об ошибках обмена данными сообщается только на выходах ERROR и STATUS инструкции USS_Port_Scan.
STATUS	OUT	Word	STATUS показывает результат запроса на чтение. Дополнительная информация для отдельных кодов состояния доступна в переменной "USS_Extended_Error".

[1] Бит DONE указывает на то, что действительные данные были считаны из адресованного электропривода и отправлены CPU. Он не указывает на то, что библиотека USS в состоянии немедленно считать следующий параметр. На электропривод необходимо отправить пустой запрос PKW и квитировать его инструкцией, прежде чем канал параметров освободится для использования соответствующим приводом. Немедленный вызов FC USS_Read_Param или USS_Write_Param для специального привода двигателя приведет к ошибке 0x818A.

13.4.3.4 Инструкция USS_Write_Param (изменить параметры в приводе)

Примечание

EEPROM инструкции записи (для EEPROM в приводе USS)

Не использовать большое количество операций записи в EEPROM. Количество операций записи в EEPROM должно быть минимальным, чтобы продлить срок службы EEPROM.

Таблица 13-49 Инструкция USS_Write_Param

LAD/FBD	SCL	Описание

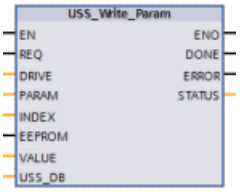
	`USS_Write_Param(REQ:=_bool_in_,`
	`DRIVE:=_usint_in_,`
	`PARAM:=_uint_in_,`
	`INDEX:=_uint_in_,`
	`EEPROM:=_bool_in_,`
	`VALUE:=_variant_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`USS_DB:=_fbtref_inout_);`
		Инструкция USS_Write_Param изменяет параметр в приводе. Все функции USS, присвоенные сети USS и коммуникационному порту PtP, должны использовать один блок данных.
		USS_Write_Param должна вызываться из OB цикла главной программы.

Таблица 13-50 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Запрос на передачу: Если REQ имеет значение TRUE, требуется новый запрос на запись. Он игнорируется, если уже есть активный запрос для этого параметра.
DRIVE	IN	USInt	Адрес привода: DRIVE является адресом привода USS. Действительный диапазон находится между приводом 1 и приводом 16.

Параметр и тип данных		Тип данных	Описание
PARAM	IN	UInt	Номер параметра: PARAM указывает, какой параметр привода записывается. Диапазон для этого параметра составляет от 0 до 2047. В некоторых приводах старший значащий байт может получить доступ к значениям PARAM, больше, чем 2047. Дополнительная информация о доступе к расширенному диапазону находится в руководстве по эксплуатации привода.
INDEX	IN	UInt	Индекс параметра: INDEX указывает, в какой индекс параметра привода должна выполняться запись. Речь идет о 16-битном значении, в котором младший байт является фактическим значением индекса, с диапазоном (от 0 до 255). Старший байт также может использоваться приводом и зависит от типа привода. Дополнительная информация находится в руководстве по эксплуатации привода.
EEPROM	IN	Bool	Сохранение в EEPROM привода: Если TRUE, транзакция параметра для записи сохраняется в привод в EEPROM привода. Если FALSE, записанное значение сохраняется временно и утрачивается при следующем включении привода.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDInt, Real	Значение параметра, в который будет выполняться запись. Оно должно быть действительным при смене состояния REQ.
USS_DB	INOUT	USS_BASE	Имя блок данных экземпляра, который создается и инициализируется при вставке инструкции USS_Drive_Control в программу.
DONE[1]	OUT	Bool	Если TRUE, DONE указывает на то, что вход VALUE был записан в привод. Этот бит устанавливается, если инструкция USS_Drive_Control распознает ответ записи привода. Этот бит сбрасывается, когда запрашиваются ответные данные, используя другой запрос USS_Drive_Control, либо при втором из следующих двух вызовов USS_Drive_Control.
ERROR	OUT	Bool	Если TRUE, ERROR указывает на ошибку и действительный выход STATUS. Все остальные выходы при возникновении ошибки устанавливаются на ноль. Об ошибках обмена данными сообщается только на выходах ERROR и STATUS инструкции USS_Port_Scan.
STATUS	OUT	Word	STATUS показывает результат запроса записи. Дополнительная информация для отдельных кодов состояния доступна в переменной "USS_Extended_Error".

[1] Бит DONE указывает на то, что действительные данные были считаны из адресованного электропривода и отправлены CPU. Он не указывает на то, что библиотека USS в состоянии немедленно считать следующий параметр. На электропривод необходимо отправить пустой запрос PKW и квитировать его инструкцией, прежде чем канал параметров освободится для использования соответствующим приводом. Немедленный вызов FC USS_Read_Param или USS_Write_Param для специального привода двигателя приведет к ошибке 0x818A.

13.4.4 Коды состояния USS

Коды состояния USS-инструкций возвращаются на выходе STATUS для функций USS.

Таблица 13-51 STATUS-коды [1]

STATUS (W#16#....)	Описание
0000	Ошибки отсутствуют
8180	Длина ответа привода не соответствовала символам, принятым от привода. Номер привода, в котором возникла ошибка, возвращается в переменной "USS_Extended_Error". Описание расширенных ошибок дается под этой таблицей.
8181	Параметр VALUE не относится к типу данных Word, Real или DWord.
8182	Пользователь ввел значение параметра типа Word, а ответ от привода получен в формате DWord или Real.
8183	Пользователь ввел значение параметра типа DWord или Real, а ответ от привода получен в формате Word.
8184	В ответной телеграмме привода была неверная контрольная сумма. Номер привода, в котором возникла ошибка, возвращается в переменной "USS_Extended_Error". Описание расширенных ошибок дается под этой таблицей.
8185	Недопустимый адрес привода (действительный диапазон адресов для привода: от 1 до16)
8186	Заданное значение скорости вне действительного диапазона (действительный диапазон заданного значения для скорости: от -200 % до 200 %).
8187	Неверный номер привода ответил на отправленный запрос. Номер привода, в котором возникла ошибка, возвращается в переменной "USS_Extended_Error". Описание расширенных ошибок дается под этой таблицей.
8188	Недопустимая длина PZD в словах (действительный диапазон = 2, 4, 6 или 8 слов)
8189	Указана недопустимая скорость передачи данных
818A	Канал запроса для параметра используется другим запросом для этого привода.
818B	Привод не отреагировал на запросы и повторы. Номер привода, в котором возникла ошибка, возвращается в переменной "USS_Extended_Error". Описание расширенных ошибок дается под этой таблицей.
818C	Привод возвратил расширенную ошибку на запрос параметра. Описание расширенных ошибок дается под этой таблицей.
818D	Привод возвратил ошибку "Несанкционированный доступ" на запрос параметра. Дополнительную информацию о том, как можно ограничить доступ к параметру, см. в руководстве по эксплуатации своего привода.
818E	Привод не был инициализирован. Этот код ошибки возвращается в USS_Read_Param или USS_Write_Param, если инструкция USS_Drive_Control не была вызвана для этого привода хотя бы один раз. Таким образом исключается ситуация, когда инициализация в первом цикле USS_Drive_Control заменит существующий запрос на чтение или запись параметров, так как при этом привод инициализируется как новая запись. Для устранения этой ошибки следует вызвать инструкцию USS_Drive_Control для данного привода.
80Ax-80Fx	Определенные ошибки возвращены из FB для PtP-коммуникаций, вызванных USS-библиотекой. - Эти коды ошибок не изменяются USS-библиотекой и определены в описаниях PtP-инструкции.

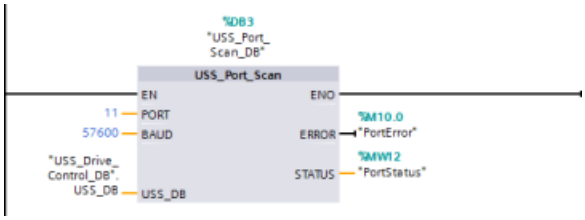
[1] Наряду с перечисленными выше ошибками USS инструкций, базовые PtP коммуникационные инструкции (Страница 1139) также могут возвращать ошибки.

Для различных кодов STATUS дополнительная информация предоставляется в переменной "USS_Extended_Error" экземплярного блока данных USS_Drive_Control. Для

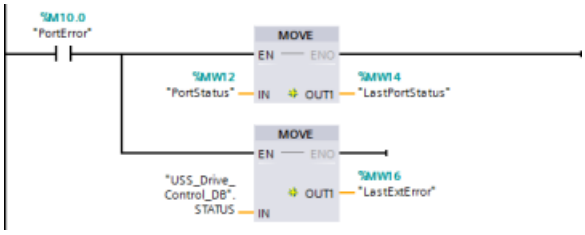
шестн. кодов параметра STATUS 8180, 8184, 8187 и 818B, переменнаяUSS_Extended_Error содержит номер привода, на котором возникла ошибка обмена данными. Для шестн. кода параметра STATUS 818C, переменная USS_Extended_Error содержит код ошибки привода, возвращаемый приводом при использовании инструкции USS_Read_Param или USS_Write_Param.

Пример: Оповещение об ошибках коммуникации

Об ошибках коммуникации (STATUS = 16#818B) сообщает только инструкция USS_Port_Scan, но не инструкция USS_Drive_Control. Пример: Если сегмент должным образом не завершен, то привод может перейти в RUN, но инструкция USS_Drive_Control покажет "0" для всех выходных параметров. В этом случае ошибку коммуникации можно обнаружить только с помощью инструкции USS_Port_Scan. Так как эта ошибка отображается только один цикл, необходимо будет добавить соответствующую логику захвата. Процесс показа в примере ниже. В этом примере, если бит ошибки инструкции USS_Port_Scan имеет значение TRUE, то значения STATUS и USS_Extended_Error помещаются в M область памяти. Номер привода помещается в переменную USS_Extended_Error, если значение кода STATUSравно шестнадцатеричным 8180, 8184, 8187 или 818B.



Сегмент 1 Состояние порта "PortStatus" и расширенные значения кодов ошибок "USS_Drive_Control_DB".USS_Extended_Error действительны только в течение одного цикла программы. Значения должны быть зарегистрированы для дальнейшей обработки.



Сегмент 2 Контакт "PortError" запускает сохранение значения "PortStatus" в "LastPortStatus" и значения "USS_Drive_Control_DB".USS_Extended_Error в "LastExtError".

Доступ для чтения и записи к внутренним параметрам привода

Приводы USS поддерживают доступ по чтению и записи к своим внутренним параметрам. Эта функция позволяет выполнять удаленное управление и конфигурирование привода. Операции доступа к параметрам привода могут перестать работать из-за ошибок, таких как выход значения из диапазона или недопустимых запросов в текущем режиме работы привода. Привод генерирует код ошибки, который возвращается в переменной "USS_Extended_Error". Этот код ошибки действует только для последнего выполнения инструкции USS_Read_Param или USS_Write_Param. Код ошибки привода сохраняется в переменной "USS_Extended_Error", если шестн. значение STATUS code равно 818C. Значение кода ошибки USS_Extended_Error зависит от модели привода. Описание расширенных кодов ошибок для функций чтения и записи для параметров см. в руководстве по эксплуатации привода.

13.4.5 Общие требования USS по настройке привода

Общие требования USS по настройке привода состоят из следующих пунктов:

- Для приводов должно быть настроено использование четырех слов PKW.
- Приводы могут конфигурироваться для 2, 4, 6 или 8 слов PZD.
- Количество слов PZD в приводе должно соответствовать входу PZD_LEN инструкции USS_Drive_Control привода.
- Скорость передачи всех приводов должна соответствовать входу BAUD инструкции USS_Port_Scan.
- Привод должен быть настроен для дистанционного управления.
- Привод должен быть настроен для уставки частоты по USS на COM канале.
- Адрес привода должен быть установлен в диапазоне 1 - 16. Этот адрес должен соответствовать входу DRIVE на блоке USS_Drive_Control привода.
- Для управления направлением привода необходимо настроить использование полярности заданного значения привода.
- Оконечная нагрузка сети RS485 должна быть включена надлежащим образом.

13.4.6 Пример: Общее USS-подключение и настройка привода

Подключение привода MicroMaster

Информация о приводах SIEMENS MicroMaster используется в качестве примера. Инструкцию по настройке других приводов можно найти в руководстве на привод.

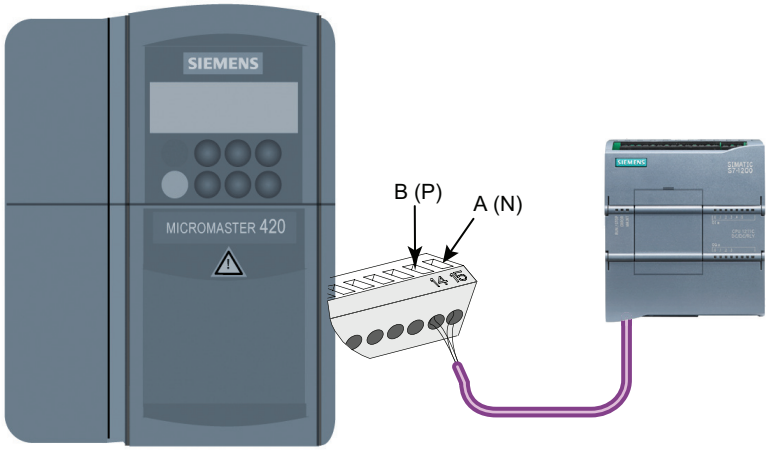
Для подключения привода MicroMaster серии 4 (MM4), вставить концы кабеля RS485 в оба безвинтовых зажима для режима USS. Стандартный кабель PROFIBUS и соединители могут использоваться для подключения S7-1200 к приводу MicroMaster.

⚠ ОСТОРОЖНО

Взаимосвязанное оборудование с различными опорными потенциалами может стать причиной протекания нежелательных токов через соединительный кабель

Эти нежелательные токи могут стать причиной ошибок обмена данными или повредить устройства. Следует убедиться, что все компоненты оборудования, которые соединяются друг с другом коммуникационным кабелем, либо имеют общий нулевой провод в цепи тока, либо выполнены с электрической изоляцией для предотвращения нежелательных электрических токов. Экран должен быть присоединен к корпусной земле или контакту 1 на 9-контактном соединителе. Рекомендуется подключить клемму 2-0 В на приводе MicroMaster к корпусной земле.

Два провода на противоположном конце кабеля RS485 должны быть вставлены в терминальный блок привода MM4. Чтобы выполнить кабельное соединение на приводе MM4, снять крышку(-и) привода, чтобы получить доступ к терминальным блокам. См. руководство пользователя MM4 для получения дополнительной информации о том, как снять крышку(-и) конкретного привода.

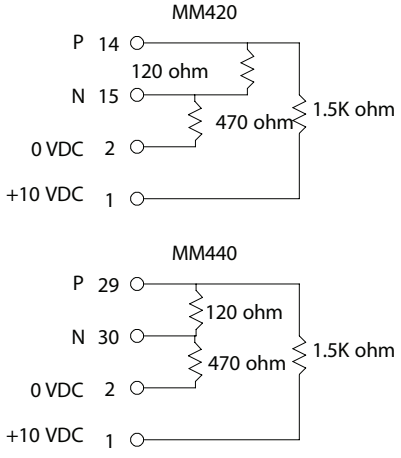


Соединения на терминальном блоке имеют цифровую маркировку. Используя соединитель PROFIBUS на стороне S7-1200, подключить клемму A кабеля к клемме 15 привода (для MM420) или клемме 30 привода (для MM440). Подключить клемму B (P) A (N) кабельного соединителя к клемме 14 привода (для MM420) или клемме 29 привода (для MM440).

Если S7-1200 является оконечным узлом в сети, или для соединения "точка-точка", необходимо использовать клеммы A1 и B1 (не A2 и B2) соединителя, так как они позволяют установить оконечную нагрузку (например, с использованием DP-соединителя типа 6ES7972-0BA40-0X40).

⚠ ОСТОРОЖНО
Установить крышки привода должным образом перед подачей питания
Убедиться, что крышки привода установлены должным образом, перед подачей питания на устройство.

Если привод сконфигурирован как оконечный узел в сети, то концевые сопротивления также должны также быть присоединены к соответствующим клеммам. На этом изображении показаны примеры подключения привода MM4 в роли последнего узла сети.



Настройка привода MM4

Перед подключением привода к S7-1200, необходимо убедиться в наличии на приводе следующих системных параметров. Настройка параметров осуществляется через клавиатуру привода:

1. Сбросьте привод на заводские настройки (по желанию).	P0010 = 30 P0970 = 1
Если пропустить шаг 1, необходимо проследить за тем, чтобы для этих параметров были установлены указанные значения:	USS PZD-длина = P2012 индекс 0 = (2, 4, 6 или 8) USS PKW-длина = P2013 индекс 0 = 4
2. Разрешить доступ по чтению и записи для всех параметров (экспертный режим).	P0003 = 3
3. Проверите параметры двигателя привода. Параметры зависят от соответствующего двигателя. Для настройки параметров P304, P305, P307, P310 и P311, необходимо сначала установить параметр P010 на 1 (режим для быстрого ввода в эксплуатацию). После настройки параметров, установить параметр P010 на 0. Параметры P304, P305, P307, P310 и P311 могут быть изменены только в режиме для быстрого ввода в эксплуатацию.	P0304 = ном. напряжение двигателя (В) P0305 = ном. ток двигателя (А) P0307 = ном. мощность двигателя (Вт) P0310 = ном. частота двигателя (Гц) P0311 = ном скорость двигателя
4. Установить локальный/дистанционный режим.	P0700 индекс 0 = 5
5. Установить заданное значение частоты в соединении COM на USS.	P1000 индекс 0 = 5
6. Время разгона (по желанию) Время в секундах, которое необходимо двигателю для ускорения до максимальной частоты.	P1120 = (от 0 до 650,00)

7. Время инерционного выбега (по желанию) Время в секундах, которое необходимо двигателю, чтобы выполнить торможение до полного останова.	P1121 = (от 0 до 650,00)
8. Установить опорную частоту последовательного соединения:	P2000 = (от 1 до 650 Гц)
9. Установить нормализацию USS:	P2009 индекс 0 = 0
10. Установить скорость в бодах для последовательного интерфейса RS485:	P2010 индекс 0 = 4 (2400 бод) 5 (4800 бод) 6 (9600 бод) 7 (19200 бод 8 (38400 бод) 9 (57600 бод) 12 (115200 бод)
11. Ввести адрес ведомого устройства. Каждый привод (макс. 31) может работать через шину.	P2011 индекс 0 = (от 0 до 31)
12. Установить значение тайм-аута для последовательного соединения. Это значение является максимально допустимым промежутком времени между двумя входящими датаграммами. Эта функция отключает инвертор в случае сбоя коммуникации. Время измеряется после получения действительной телеграммы. Если в течение указанного времени не принимается следующая датаграмма, инвертор отключается и показывает код ошибки F0070. Установка значения на ноль приводит к отключению управления.	P2014 индекс 0 = (от 0 до 65.535 мс) 0 = тайм-аут выключен
13. Выполнить передачу данных из RAM в EEPROM:	P0971 = 1 (начать передачу). Сохранить настройки параметров в EEPROM.

13.5 Коммуникация Modbus

13.5.1 Обзор коммуникация с помощью MODBUS RTU и MODBUS TCP

Функциональные коды Modbus

- CPU, работающий в качестве ведущего устройства Modbus RTU (или клиента Modbus TCP), может читать/записывать как данные, так и состояния ввода-вывода в удаленном ведомом устройстве Modbus RTU (или на сервере Modbus TCP). Удаленные данные могут быть считаны и затем обработаны в программной логике.
- CPU, работающий в качестве ведомого устройства Modbus RTU (или сервера Modbus TCP), позволяет устройству верхнего уровня читать/записывать как данные, так и состояния ввода-вывода в памяти CPU. Ведущее устройство RTU (или клиент Modbus TCP) может записывать новые значения в память CPU ведомого устройства/сервера, которая доступна для программной логики.

⚠ ПРЕДУПРЕЖДЕНИЕ

Если злоумышленник сможет получить физический доступ к сетям пользователя, то он сможет читать и записывать данные.

TIA Portal, CPU и устройства HMI (кроме устройств HMI, работающих с GET / PUT) используют безопасную связь с защитой от взломов путём замещения оригинала и атак через посредника. После активации коммуникации, обмен подписанными сообщениями происходит в форме открытого текста, который позволяет злоумышленнику читать данные, но защищает от их несанкционированной записи. TIA Portal, а не процесс коммуникации, шифрует данные блоков с защитой ноу-хау.

Все другие формы коммуникации (обмен I/O данным через PROFIBUS, PROFINET, AS-i или другую шину ввода-вывода, GET / PUT, T-Block и коммуникационные модули (CM)) не имеют функций обеспечения безопасности. Эти типы коммуникации должны быть защищены путем ограничения физического доступа. Если злоумышленник сможет получить физический доступ к сетям через эти типы коммуникации, то он сможет при необходимости читать и записывать данные.

Для получения информации и рекомендаций по обеспечению безопасности см. Operational Guidelines for Industrial Security (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf).

Таблица 13-52 Функции для чтения данных: Чтение удаленных входов/выходов и данных программы

Код функции Modbus	Функции для чтения ведомого устройства (сервера) - стандартная адресация
01	Считывание выходных битов: от 1 до 2000 бит за вызов
02	Считывание входных битов: от 1 до 2000 бит за вызов
03	Считывание регистров хранения: от 1 до 125 бит за вызов
04	Считывание входных слов: от 1 до 125 бит за вызов

Таблица 13-53 Функции для записи данных: Запись удаленных входов/выходов и изменение данных программы

Код функции Modbus	Функции для записи на ведомое устройство (сервер) - стандартная адресация
05	Запись выходного бита: 1 бит на запрос
06	Запись регистра хранения: 1 слово на запрос
15	Запись одного или нескольких выходных бит: от 1 до 1968 бит за вызов
16	Запись одного или нескольких регистров хранения: от 1 до 123 бит за вызов

- Коды функций Modbus 08 и 11 предоставляют диагностическую информацию о коммуникации ведомого устройства.
- Код Modbus функции 0 рассылает сообщение всем ведомым устройствам (без ответа ведомого устройства). Функция широковещательной передачи недоступна для Modbus TCP, так как коммуникация основана на установлении соединения.

Таблица 13-54 Адреса станции в сети Modbus

Станция		**Адрес**
Станция RTU	Стандартный адрес станции	От 1 до 247
	Расширенный адрес станции	От 1 до 65535
Станция TCP	Адрес станции	IP-адрес и номер порта

Адреса памяти Modbus

Фактическое количество доступных адресов памяти Modbus зависит от модели CPU, объема рабочей памяти и процента использования памяти CPU другими программными данными. Таблица ниже приводит номинальное значение диапазона адресов.

Таблица 13-55 Адреса памяти Modbus

Станция		**Диапазон адресов**
Станция RTU	Адрес в стандартной памяти	10 K
	Адрес в расширенной памяти	64 K
Станция TCP	Адрес в стандартной памяти	10 K

Коммуникация Modbus RTU

Modbus RTU (Remote Terminal Unit, удаленный протокол) представляет собой стандартный протокол для коммуникации в сети и использует электрические соединения RS232 или RS485 для последовательной передачи данных между устройствами Modbus в сети. Можно добавлять PtP сетевые порты (точка-точка) в CPU с помощью RS232 или RS485 CM или RS485 CB.

Modbus RTU использует сеть ведущий/ведомый, где вся коммуникация инициируется одним единственным ведущим устройством, а ведомые устройства могут только реагировать на запросы ведущего устройства. Ведущее устройство отправляет запрос на адрес ведомого устройства, и только этот адрес ведомого устройства отвечает на команду.

Коммуникация Modbus TCP

Modbus TCP (Transmission Control Protocol, протокол управления передачей) является стандартным протоколом сетевых коммуникаций, который использует соединитель PROFINET в CPU для TCP/IP коммуникаций. Никакой дополнительный коммуникационный аппаратный модуль не требуется.

Modbus TCP использует соединения Open User Communications (OUC) в качестве канала связи Modbus. Несколько соединений клиент-сервер могут существовать, в дополнение к соединению между STEP 7 и CPU. Смешанные соединения клиента и сервера поддерживаются до максимального количества соединений, допустимых для данной модели CPU (Страница 807).

Каждое соединение MB_SERVER должно использовать уникальный экземплярный DB и номер IP порта. Для каждого IP порта поддерживается только одно соединение. Каждый MB_SERVER (с его уникальным экземплярным DB и IP портом) должен быть выполнен индивидуально для каждого соединения.

Клиент Modbus TCP (ведущее устройство) должен управлять соединением клиент-сервер с помощью параметра DISCONNECT. Базовые операции Modbus клиента показаны ниже.

1. Установить соединение с определенным сервером (ведомое устройство) по IP-адресу и номеру IP-порта
2. Запустить клиентскую передачу Modbus сообщений и получить ответы от сервера
3. При необходимости инициировать разъединение клиента и сервера, чтобы выполнить соединение с другим сервером

Инструкции Modbus RTU в программе

- Modbus_Comm_Load: Выполнение инструкции Modbus_Comm_Load необходимо для настройки таких параметров PtP, как скорость передачи данных, четность и контроль потока. После того как была выполнена конфигурация порта CPU для протокола Modbus RTU, он может быть использован инструкцией Modbus_Master или инструкцией Modbus_Slave.
- Modbus_Master: При помощи инструкции Modbus_Master CPU может использоваться в качестве ведущего устройства Modbus-RTU для коммуникации с одним или несколькими ведомыми устройствами Modbus.
- Modbus_Slave: При помощи инструкции Modbus_Slave CPU может использоваться в качестве ведомого устройства Modbus-RTU для коммуникации с ведущим устройством Modbus.

Инструкции Modbus TCP в программе

- MB_CLIENT: Выполняет TCP соединение клиент-сервер, отправляет сообщение с командой, принимает ответ и управляет разъединением с сервером.
- MB_SERVER: Соединяет с Modbus TCP клиентом по запросу, получает Modbus сообщение и отправляет ответ.

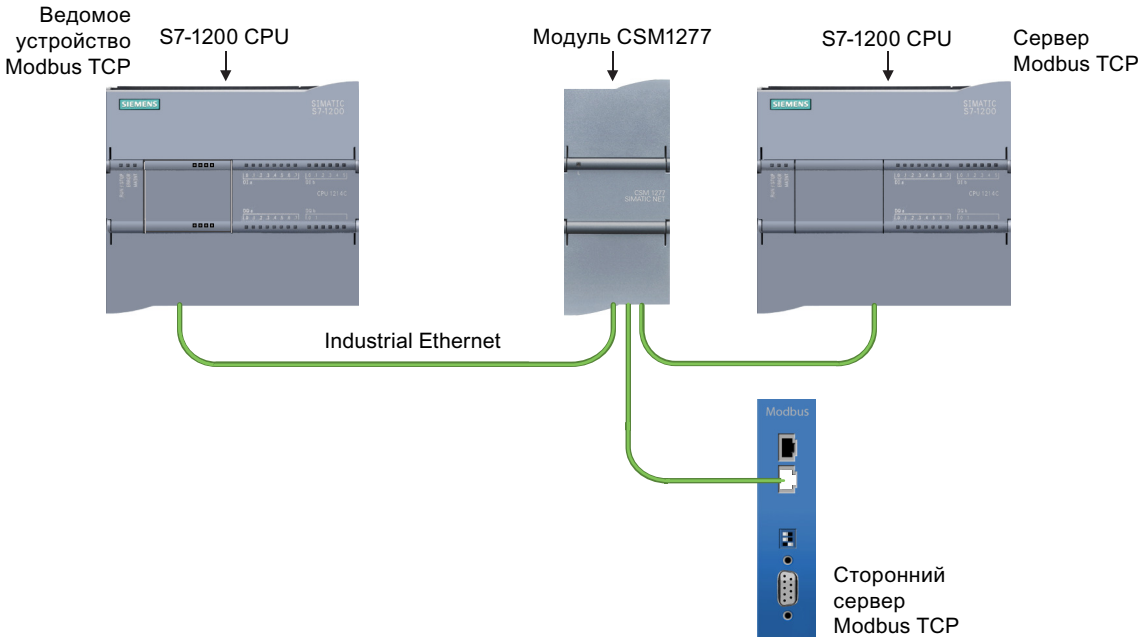
13.5.2 Modbus TCP

13.5.2.1 Обзор

От версии V4.1 модуля S7-1200 CPU в комбинации со STEP 7 V13 SP1 расширяются возможности Modbus TCP в части использования T-блок инструкций.

Поэтому S7-1200 поддерживает два набора PtP-инструкций:

- Старые инструкции Modbus TCP (Страница 1329): Эти Modbus TCP инструкции существовали до версии V4.0 для S7-1200.
- Инструкции Modbus TCP (Страница 1200): Эти инструкции Modbus TCP обеспечивают всю функциональность старых инструкций.



13.5.2.2 Выбор версии Modbus TCP инструкций

Следующие версии инструкций Modbus TCP доступны в STEP 7:

- Старая версия 2.1: совместима со всеми моделями CPU и CM
- Старая версия 3.1: совместима со всеми моделями CPU и CM
- Версия 4.2: совместима с CPU от версии V4.0 и CM от версии V2.1
- Версия 5.1: совместима с CPU от версии V4.2 и CM от версии V2.1

Учитывая совместимость и для упрощения миграции можно выбирать, какая версия инструкций будет вставлена в программу пользователя.

Вызвать на карте задач "Инструкция" инструкции Modbus TCP в группе Коммуникационный процессор в пункте "Другие".

Для изменения версии инструкции Modbus TCP, выбрать соответствующую версию в раскрывающемся списке. Можно выбрать группу или отдельные инструкции.



Если инструкция Modbus TCP размещается через дерево инструкций в программе, то в дереве проекта создается новый экземпляр FB. Для просмотра нового экземпляра FB

открыть в дереве проекта PLC_x > Программные блоки > Системные блоки > Программные ресурсы.

Чтобы узнать версию инструкции Modbus TCP в программе, следует вызвать свойства в дереве проекта, а не свойства блочного элемента на экране в редакторе текстов программ. Выбрать в дереве проекта экземпляр FB инструкции Modbus TCP, кликнуть правой кнопкой мыши, выбрать "Свойства" и после страницу "Информация", чтобы отобразить номер версии инструкции Modbus TCP.

13.5.2.3 Инструкции Modbus TCP

Инструкция MB_CLIENT (обмен данными с использованием PROFINET в качестве клиента Modbus TCP)

Таблица 13-56 Инструкция MB_CLIENT

LAD/FBD	SCL	Описание

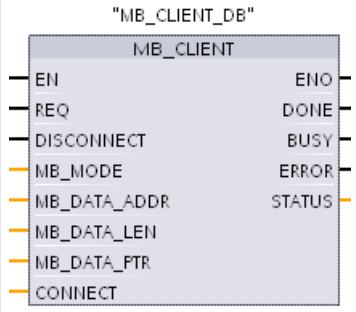
	`"MB_CLIENT_DB"(`
	`REQ:=_bool_in_,`
	`DISCONNECT:=_bool_in_,`
	`MB_MODE:=_usint_in_,`
	`MB_DATA_ADDR:=_udint_in_,`
	`MB_DATA_LEN:=_uint_in_,`
	`DONE=>_bool_out_,`
	`BUSY=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`MB_DATA_PTR:=_variant_inout_,`
	`CONNECT:=_variant_inout_);`
		MB_CLIENT выполняет обмен данными как Modbus TCP клиент через порт PROFINET на S7-1200 CPU. Никакой дополнительный коммуникационный аппаратный модуль не требуется.
		MB_CLIENT может выполнить соединение клиент-сервер, отправить запрос функции Modbus, получить ответ и управлять разъединением от сервера Modbus TCP.

Таблица 13-57 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	In	Bool	FALSE = Нет коммуникационного запроса Modbus TRUE = Запрос на обмен данными с сервером Modbus TCP
DISCONNECT	IN	Bool	Параметр DISCONNECT позволяет программе управлять соединением с сервером Modbus. Если DISCONNECT = 0 и соединение не существует, то MB_CLIENT пытается выполнить соединение с назначенным IP-адресом и номером порта. Если DISCONNECT = 1 и соединение существует, то предпринимается попытка отключения. Пока этот вход активирован, никакая другая операция не будет предпринята.
MB_MODE	IN	USInt	Выбор режима: Назначает тип запроса (чтение, запись или диагностика). Дополнительную информацию можно найти в таблице функций Modbus ниже.

Параметр и тип данных		Тип данных	Описание
MB_DATA_ADDR	IN	UDInt	Начальный адрес Modbus: Назначает начальный адрес данных, к которым должен получить доступ MB_CLIENT. Действительные адреса указаны в таблице функций Modbus ниже.
MB_DATA_LEN	IN	UInt	Длина данных Modbus: Назначает количество бит или слов, к которым будет обращаться этот запрос. Действительные длины указаны в таблице функций Modbus ниже.
MB_DATA_PTR	IN_OUT	Variant	Указатель на регистр данных Modbus: Регистр помещает в буфер данные, идущие в или приходящие от сервера Modbus. Указатель должен назначать не оптимизированный глобальный DB или адрес в M памяти.
CONNECT	IN_OUT	Variant	Ссылка на структуру блока данных, которая содержит параметры соединения в системном типе данных "TCON_IP_v4".
DONE	OUT	Bool	Бит DONE = TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок.
BUSY	OUT	Bool	• 0 - нет обрабатываемых инструкций MB_CLIENT • 1 - обрабатывается инструкция MB_CLIENT
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение TRUE, после того, как последнее выполнение MB_CLIENT было завершен с ошибкой. Код ошибки в параметре STATUS действителен только в цикле, где ERROR = TRUE.
STATUS	OUT	Word	Условие выполнения

Примечание

Требования к версии прошивки CPU

Инструкции Modbus TCP, описанные в этом разделе руководства, требуют прошивки версии V4.1 или выше.

Параметр REQ

FALSE = Нет коммуникационного запроса Modbus
TRUE = Запрос на обмен данными с сервером Modbus TCP

Если никакой из экземпляров MB_CLIENT не активен и параметр DISCONNECT=0, то при REQ=1 запускается новый запрос Modbus. Если соединение еще не установлено, то устанавливается новое соединение.

Если тот же экземпляр MB_CLIENT будет выполняться снова с DISCONNECT=0 и REQ=1 перед завершением текущего запроса, то последующая передача Modbus не будет выполнена. Однако, как только текущий запрос завершен, новый запрос может быть обработан, если MB_CLIENT выполняется с REQ=1.

Если текущий коммуникационный запрос MB_CLIENT завершен, бит DONE устанавливается в TRUE на один цикл. Бит DONE может использоваться в качестве временного окна для упорядочения множественных запросов MB_CLIENT.

Примечание

Согласованность входных данных во время обработки MB_CLIENT

Как только Modbus клиент инициирует Modbus операцию, все входные состояния сохраняются в системе и затем сравниваются при каждом последующем вызове. Сравнение используется, чтобы определить, был ли этот определенный вызов инициатором активного клиентского запроса. Можно выполнить несколько вызовов MB_CLIENT , используя общий экземплярный блок данных.

Важно, чтобы входы не были изменены в течение промежутка времени, когда операция MB_CLIENT активно обрабатывается. Если это правило не соблюдается, то MB_CLIENT не может определить активный экземпляр.

Параметры MB_MODE и MB_DATA_ADDR выбирают коммуникационные функции Modbus

Инструкция MB_CLIENT использует вход MB_MODE вместо входа кода функции. MB_DATA_ADDR назначает стартовый адрес удаленных данных Modbus.

Комбинация из MB_MODE и MB_DATA_ADDR определяет код функции, используемый в собственной телеграмме Modbus. В нижеследующей таблице показана взаимосвязь между параметром MB_MODE, MB_DATA_ADDR и функцией Modbus.

Таблица 13-58 Функции Modbus

MB_MODE	Функция Modbus	Длина данных	Операция и данные	MB_DATA_ADDR
0	01	От 1 до 2000	Чтение выходных битов: От 1 до 2000 бит за запрос	От 1 до 9999
101	01	От 1 до 2000	Чтение выходных битов: От 1 до 2000 бит за запрос	От 00000 до 65535
0	02	От 1 до 2000	Чтение входных битов: От 1 до 2000 бит за запрос	От 10001 до 19999
102	02	От 1 до 2000	Чтение входных битов: От 1 до 2000 бит за запрос	От 00000 до 65535
0	03	От 1 до 125	Чтение регистра хранения: От 1 до 125 слов за запрос	От 40001 до 49999 или от 400001 до 465535
103	03	От 1 до 125	Чтение регистра хранения: От 1 до 125 слов за запрос	От 00000 до 65535
0	04	От 1 до 125	Чтение входных слов: От 1 до 125 слов за запрос	От 30001 до 39999
104	04	От 1 до 125	Чтение входных слов: От 1 до 125 слов за запрос	От 00000 до 65535
1	05	1	Запись выходного бита: Один бит на запрос	От 1 до 9999

MB_MODE	Функция Modbus	Длина данных	Операция и данные	MB_DATA_ADDR
105	05	1	Запись выходного бита: Один бит на запрос	От 00000 до 65535
1	06	1	Запись регистра хранения: 1 слово на запрос	От 40001 до 49999 или от 400001 до 465535
106	06	1	Запись регистра хранения: 1 слово на запрос	От 00000 до 65535
1	15	От 2 до 1968	Запись нескольких выходных битов: От 2 до 1968 бит за запрос	От 1 до 9999
1	16	От 2 до 123	Запись нескольких регистров хранения: От 2 до 123 слов за запрос	От 40001 до 49999 или от 400001 до 465535
2	15	От 1 до 1968	Запись одного или нескольких выходных битов: От 1 до 1968 бит за запрос	От 1 до 9999
2	15	От 1 до 1968	Запись одного или нескольких выходных битов: От 1 до 1968 бит за запрос	От 1 до 9999
2	15	От 1 до 1968	Запись одного или нескольких выходных битов: От 1 до 1968 бит за запрос	От 1 до 9999
115	15	От 1 до 1968	Запись одного или нескольких выходных битов: От 1 до 1968 бит за запрос	От 00000 до 65535
2	16	От 1 до 123	Запись одного или нескольких регистров хранения: От 1 до 123 слов за запрос	От 40001 до 49999 или от 400001 до 465535
116	16	От 1 до 123	Запись одного или нескольких регистров хранения: От 1 до 123 слов за запрос	От 00000 до 65535
11	11	0	Считывание слова состояния и счетчика событий коммуникации с сервером. Слово состояния указывает занято/не занято (0 = не занято, 0xFFFF = занято). Счетчик событий инкрементируется при каждой удачной обработке сообщения. Как параметр MB_DATA_ADDR, так и параметр MB_DATA_LEN из MB_CLIENT игнорируются для этой функции.	
80	08	1	Проверить состояние сервера с помощью диагностического кода 0x0000 (петлевой тест, эхо запроса от сервера) 1 слово на запрос	

MB_MODE	Функция Modbus	Длина данных	Операция и данные	MB_DATA_ADDR
81	08	1	Сбросить счетчик событий сервера с помощью диагностического кода 0x000A 1 слово на запрос	
3 до 10, 12 до 79, 82 до 100, 107 до 114, 117 до 255			Зарезервировано	

Примечание

MB_DATA_PTR назначает буфер для хранения данных чтения/записи в Modbus TCP сервер

Буфер данных может быть расположен в не оптимизированном глобальном DB или M области памяти.

Для буфера в M памяти следует использовать формат указателя Any. Это формат P# "Адрес бита" "Тип данных" "Длина". Пример: P#M1000.0 WORD 500.

Параметр MB_DATA_PTR назначает коммуникационный буфер

- Функции коммуникации MB_CLIENT:
  - Читать и записывать 1-битовые данные из адресов сервера Modbus (от 00001 до 09999)
  - Читать и записывать 1-битовые данные из адресов сервера Modbus (от 10001 до 19999)
  - Читать 16-разрядные данные слов из адресов сервера Modbus (от 30001 до 39999) и (от 40001 до 49999)
  - Записывать 16-разрядные данные слов в адреса сервера Modbus (от 40001 до 49999)
- Данные размерностью слово или бит передаются в/из DB или M буфера памяти, назначенного MB_DATA_PTR.
- Если DB назначен, как буфер MB_DATA_PTR, то необходимо присвоить типы данных всем элементам данных DB.
  - 1-битовый тип данных Bool представляет один Modbus адрес бита
  - 16-битовые типы данных отдельного слова такие как WORD, UInt и Int представляют один Modbus адрес слова
  - 32-разрядные типы данных двойного слова такие как DWORD, DInt и Real представляют два Modbus адреса слова

- MB_DATA_PTR могут быть назначены сложные элементы DB, такие как
  - Массивы
  - Именованные структуры, где каждый элемент уникален.
  - Именованные сложные структуры, где каждый элемент имеет уникальное имя и 16- или 32-битный типа данных.
- Не требуется, чтобы области данных MB_DATA_PTR были в том же глобальном блоке данных (или M области памяти). Можно назначить один блок данных для чтения по Modbus, другой блок данных для записи по Modbus или один блок данных для каждого MB_CLIENT.

Параметр CONNECT назначает данные для создания соединения PROFINET

Необходимо использовать глобальный блок данных и сохранить требуемые данные соединения, прежде чем можно будет сослаться на этот DB в параметре CONNECT.

1. Создать новый глобальный DB или использовать существующий глобальный DB, чтобы сохранить данные CONNECT. Можно использовать один DB, чтобы сохранить несколько структур данных TCON_IP_v4. Каждое подключение клиента или сервера Modbus TCP использует структуру данных TCON_IP_v4. В параметре CONNECT адресуются данные соединения.
2. Присвоить смысловое имя DB и статической переменной. Например, назвать блок данных "Соединения Modbus", а статическую переменную "TCPactive_1" (для Modbus TCP подключения клиента 1)
3. В редакторе DB назначить статической переменной примера "TCPactive_1" системный тип данных "TCON_IP_v4" в столбце "Тип данных".
4. Развернув структуру TCON_IP_v4, можно изменить параметры соединения, как показано на следующем изображении.
5. Изменить данные в структуре TCON_IP_v4 для соединения MB_CLIENT.
6. Ввести ссылку на структуру из DB для параметра CONNECT в MB_CLIENT. Имя ссылки в этом примере "Modbus соединения".TCPaktiv_1.

Modbus connections

	Name	Data type	Start value	Comment
1	Static			
2	TCPactive_1	TCON_IP_v4		
3	InterfaceId	HW_ANY	64	HW-identifier of IE-interface submodule
4	ID	CONN_OUC	1	connection reference / identifier
5	ConnectionType	Byte	16#0B	type of connetion: 11=TCP/IP, 19=UDP (17=TC...
6	ActiveEstablished	Bool	True	active/passive connection establishment
7	RemoteAddress	IP_V4		remote IP address (IPv4)
8	ADDR	array [1..4] of Byte		IPv4 address
9	ADDR[1]	Byte	192	
10	ADDR[2]	Byte	168	
11	ADDR[3]	Byte	2	
12	ADDR[4]	Byte	241	
13	RemotePort	UInt	502	remote UDP/TCP port number
14	LocalPort	UInt	0	local UDP/TCP port number

Изменение данных TCP_IP_v4 DB для каждого соединения MB_CLIENT

- **InterfaceID**: Кликнуть в окне конфигурации устройства по изображению PROFINET порта CPU. После кликнуть мышью по вкладке "Общие" и использовать предложенный там аппаратный идентификатор.
- **ID**: Ввести ID соединения в диапазоне от 1 до 4095. Для коммуникации Modbus TCP используются базовые инструкции TCON, TDISCON, TSEND и TRCV для открытых коммуникационных соединений пользователя (Open user communication).
- **ConnectionType**: Для TCP/IP следует использовать по умолчанию 16#0B (десятичное = 11).
- **ActiveEstablished**: Это значение должно равняться 1 или TRUE. Соединение активно, поскольку MB_CLIENT инициирует коммуникацию Modbus.
- **RemoteAddress**: Ввести IP-адрес целевого сервера Modbus TCP в четыре элемента массива ADDR. Например, ввести 192.168.2.241, как на изображении выше.
- **RemotePort**: Значение по умолчанию равно 502. Это число представляет номер IP порта для сервера Modbus, к которому MB_CLIENT пытается подключиться для коммуникации. Некоторые серверы Modbus сторонних производителей требуют, чтобы использовался другой номер порта.
- **LocalPort**: Это значение для соединения MB_CLIENT должно быть 0.

Подключение нескольких клиентов

Клиент Modbus TCP может поддерживать параллельные соединения вплоть до максимального количества соединений для открытых коммуникационных соединений пользователя (Open user communication), разрешенных для устройства PLC. Общее количество соединений для устройства PLC, включая клиентов и сервер Modbus TCP, не должно превышать максимальное количество поддерживаемых открытых коммуникационных соединений пользователя (Страница 807).

Отдельные параллельные клиентские соединения должны следовать данным правилам:

- Каждое соединение MB_CLIENT должно использовать уникальный DB экземпляра.
- Каждое соединение MB_CLIENT должно назначать уникальный IP-адрес сервера.
- Каждое соединение MB_CLIENT должно назначать уникальный ID соединения.
- Уникальные номера портов IP могут потребоваться в зависимости от конфигурации сервера.

Различные ID соединения должны использоваться с каждым экземплярным DB. Таким образом, экземплярный DB и ID соединения составляют пару и должны быть уникальными для каждого соединения.

Таблица 13-59 Экземплярный блок данных MB_CLIENT: доступные для пользователя статические переменные

Переменная	Тип данных	По умолчанию	Описание
Blocked_Proc_Timeout	Real	3.0	Время (в секундах), в течение которого ожидается заблокированный экземпляр клиента Modbus, прежде чем этот экземпляр будет удален со статусом ACTIVE. Это может произойти, например, когда был выставлен клиентский запрос и впоследствии приложение прекращает выполнять клиентскую функцию до полного завершения запроса. Максимальный предел для S7-1200 составляет 55 секунд.
MB_Unit_ID	Word	255	Идентификатор устройства Modbus: Сервер Modbus TCP адресуется с помощью своего IP-адреса. Поэтому параметр MB_UNIT_ID не используется при адресации Modbus TCP. Параметр MB_UNIT_ID соответствует полю адреса ведомого устройства в протоколе Modbus RTU. Если сервер Modbus TCP используется в качестве шлюза к протоколу Modbus RTU, MB_UNIT_ID может использоваться для идентификации ведомого устройства, подключенного к последовательной сети. Параметр MB_UNIT_ID мог бы использоваться для передачи запроса по корректному адресу ведомого устройства Modbus RTU. Некоторые устройства Modbus TCP могут потребовать, чтобы параметр MB_UNIT_ID был в ограниченном диапазоне.
RCV_TIMEOUT	Real	2,0	Время в секундах, в течение которого MB_CLIENT ожидает сервер, чтобы ответить на запрос.
Connected	Bool	0	Указывает на установление соединения с назначенным сервером: 1 = соединение установлено , 0 = соединение не установлено

Таблица 13-60 Ошибка протокола MB_CLIENT

STATUS (W#16#....)	Код ответа Modbus клиенту (B#16#)	Ошибка протокола Modbus
8381	01	Код функции не поддерживается
8382	03	Ошибка в длине данных
8383	02	Ошибка в адресе данных или доступ вне диапазона адресов MB_HOLD_REG
8384	03	Ошибка в значении данных
8385	03	Диагностический код не поддерживается (код функции 08)

Таблица 13-61 Коды состояния выполнения MB_CLIENT [1]

STATUS (W#16#....)	**Ошибка параметра MB_CLIENT**
7001	MB_CLIENT ожидает ответа сервера Modbus на запрос подключения или отключения на назначенном порте TCP. Этот код возвращается только для первого выполнения операции подключения или отключения.
7002	MB_CLIENT ожидает ответа сервера Modbus на запрос подключения или отключения на назначенном порте TCP. Этот код будет возвращен для любого последующего выполнения при ожидании завершения операции подключения или отключения.
7003	Операция отключения успешно завершена (действует только для одного цикла PLC)
80C8	Сервер не ответил в назначенное время. MB_CLIENT должен принять ответ, используя ID транзакции, который был первоначально передан в течение назначенного времени, или будет возвращена данная ошибка. Проверить соединение с серверным устройством Modbus. Эта ошибка возвращается только после того, как были предприняты повторные попытки (если применимо).
8188	Недействительный режим
8189	Недействительный адрес данных
818A	Недопустимая длина данных
818B	Недействительный указатель на область DATA_PTR Здесь речь может идти о комбинации MB_DATA_ADDRESS + MB_DATA_LEN.
818C	Указатель DATA_PTR указывает на не оптимизированною область DB (при этом речь должна идти об не оптимизированной области DB или область M памяти)
8200	Порт занят обработкой существующего запроса Modbus.
8380	Принятый кадр Modbus неправильный, или было получено слишком мало байтов.
8387	Назначенный параметр ID соединения отличается от ID, использованного для предыдущих запросов. Может быть только один ID соединения, используемый в каждом экземплярном DB для MB_CLIENT. Этот код также возвращается как внутренняя ошибка, если ID протокола Modbus TCP, полученный от сервера, не 0.
8388	Сервер Modbus возвратил количество данных, которое отличается от запрошенного количества. Этот код применяется только к функциям Modbus 15 и 16.

[1] В дополнение к упомянутым выше ошибкам MB_CLIENT, ошибки могут быть возвращены из базовых коммуникационных инструкций T блока (TCON, TDISCON, TSEND и TRCV).

Инструкция MB_SERVER (обмен данными с использованием PROFINET в качестве сервера Modbus TCP)

Таблица 13-62 Инструкция MB_SERVER

LAD/FBD	SCL	Описание

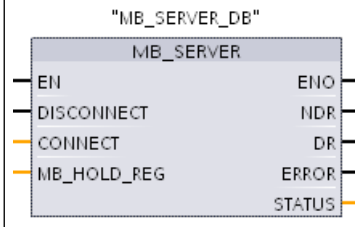
	`"MB_SERVER_DB"(`
	`  DISCONNECT:=_bool_in_,`
	`  CONNECT:=_variant_in_,`
	`  NDR=>_bool_out_,`
	`  DR=>_bool_out_,`
	`  ERROR=>_bool_out_,`
	`  STATUS=>_word_out_,`
	
	`MB_HOLD_REG:=_variant_inout_);`
		MB_SERVER выполняет обмен данными как Modbus TCP сервер через порт PROFINET на S7-1200 CPU. Никакой дополнительный коммуникационный аппаратный модуль не требуется.
		MB_SERVER может принять запрос на соединение с клиентом Modbus TCP, получить запрос функции Modbus и отправить ответное сообщение.

Таблица 13-63 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
DISCONNECT	IN	Bool	MB_SERVER пытается установить "пассивное" соединение с устройством партнера. Это означает, что сервер пассивно "прислушивается" к запросу на установление TCP соединения от любого запрашивающего IP-адреса. Если DISCONNECT = 0 и соединения отсутствуют, то может быть установлено пассивное соединение. Если DISCONNECT = 1 и соединение существует, то инициируется операция отключения. Этот параметр позволяет программе выбирать момент для подтверждения соединения. Пока этот вход активирован, никакая другая операция не будет предпринята.
CONNECT	IN	Variant	Ссылка на структуру блока данных, которая содержит параметры соединения в системном типе данных "TCON_IP_v4".
MB_HOLD_REG	IN_OUT	Variant	Указатель на Modbus регистр хранения MB_SERVER: Регистр хранения должен быть либо не оптимизированным глобальным DB, либо адресом в M памяти. Эта область памяти используется, чтобы хранить данные, к которым может обращаться Modbus клиент, используя регистровые функции: 3 (чтение), 6 (запись) и 16 (запись) и 23 (чтение/запись).
NDR	OUT	Bool	Новые данные готовы: 0 = нет новых данных, 1 = указывает на то, что новые данные были записаны клиентом Modbus
DR	OUT	Bool	Считывание данных: 0 = нет чтения данных, 1 = указывает на то, что данные были считаны клиентом Modbus
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение TRUE, после того, как выполнение MB_SERVER было завершен с ошибкой. Код ошибки в параметре STATUS действителен только в цикле, где ERROR = TRUE.
STATUS	OUT	Word	Условие выполнения

Примечание

Требования к версии прошивки CPU

Инструкции Modbus TCP, описанные в этом разделе руководства, требуют прошивки версии V4.1 или выше.

Примечание

Использование функции 23 с инструкцией MB_SERVER

Инструкция MB_SERVER поддерживает использование кода функции 23 для записи/чтения регистра хранения в одном запросе. Но инструкция MB_CLIENT НЕ поддерживает эту функцию и возвращает код ошибки. Также необходимо учитывать, что хотя инструкция и содержит обе операции, чтение и запись, но операция записи обрабатывается до операции чтения.

Параметр CONNECT назначает данные для создания соединения PROFINET

Необходимо использовать глобальный блок данных и сохранить требуемые данные соединения, прежде чем можно будет сослаться на этот DB в параметре CONNECT.

1. Создать новый глобальный DB или использовать существующий глобальный DB, чтобы сохранить данные CONNECT. Можно использовать один DB, чтобы сохранить несколько структур данных TCON_IP_v4. Каждое подключение клиента или сервера Modbus TCP использует структуру данных TCON_IP_v4. В параметре CONNECT адресуются данные соединения.
2. Присвоить смысловое имя DB и статической переменной. Например, назвать блок данных "Соединения Modbus", а статическую переменную "TCPactive_1" (для Modbus TCP подключения клиента 1)
3. В редакторе DB назначить статической переменной примера "TCPactive_1" системный тип данных "TCON_IP_v4" в столбце "Тип данных".
4. Развернув структуру TCON_IP_v4, можно изменить параметры соединения, как показано на следующем изображении.
5. Изменить данные в структуре TCON_IP_v4 для соединения MB_SERVER.
6. Ввести ссылку на структуру из DB для параметра CONNECT в MB_SERVER . Имя ссылки в этом примере "Modbus соединения".TCPpassiv_1.

Modbus connections

	Name	Data type	Start value	Comment
1	Static			
2	TCPpassive_1	TCON_IP_v4		
3	InterfaceId	HW_ANY	64	HW-identifier of IE-interface submodule
4	ID	CONN_OUC	1	connection reference / identifier
5	ConnectionType	Byte	16#0B	type of connetion: 11=TCP/IP, 19=UDP (17=TC...
6	ActiveEstablished	Bool	False	active/passive connection establishment
7	RemoteAddress	IP_V4		remote IP address (IPv4)
8	ADDR	array [1..4] of Byte		IPv4 address
9	ADDR[1]	Byte	192	
10	ADDR[2]	Byte	168	
11	ADDR[3]	Byte	2	
12	ADDR[4]	Byte	241	
13	RemotePort	UInt	0	remote UDP/TCP port number
14	LocalPort	UInt	502	local UDP/TCP port number

Изменение данных TCP_IP_v4 DB для каждого соединения MB_SERVER

- **InterfaceID**: Кликнуть в окне конфигурации устройства по изображению PROFINET порта CPU. После кликнуть мышью по вкладке "Общие" и использовать предложенный там аппаратный идентификатор.
- **ID**: Ввести однозначное для этого соединения число в диапазоне от 1 до 4095. Для коммуникации Modbus TCP используются базовые инструкции TCON, TDISCON, TSEND и TRCV для открытых коммуникационных соединений пользователя (Open user communication). Допускается до восьми одновременных открытых коммуникационных соединений пользователя.
- **ConnectionType**: Для TCP/IP следует использовать по умолчанию 16#0B (десятичное значение = 11).
- **ActiveEstablished**: Это значение должно равняться 0 или FALSE. В этом случае соединение пассивно и MB_SERVER ожидает коммуникационного запроса от клиента Modbus.
- **RemoteAddress**: Существует два варианта.
  - Использовать 0.0.0.0 и MB_CLIENT ответит на запрос по Modbus от любого TCP клиента
  - Если ввести IP-адрес целевого клиента Modbus TCP, то MB_CLIENT отвечает на запрос, приходящий только от IP-адреса этого клиента. Например, ввести 192.168.2.241, как на изображении выше.
- **RemotePort**: Это значение для соединения MB_SERVER должно быть 0.
- **LocalPort**: Значение по умолчанию равно 502. Это число представляет номер IP порта для клиента Modbus, к которому MB_SERVER пытается подключиться для коммуникации. Некоторые клиенты Modbus сторонних производителей требуют другого номера порта.

Адреса Modbus и образа процесса

MB_SERVER позволяет входящим кодам функций Modbus (1, 2, 4, 5 и 15) читать/записывать биты/слова непосредственно в образе процесса ввода/вывода. Для кодов функций передачи данных (3, 6 и 16), параметр MB_HOLD_REG должен быть определен как тип данных, длиннее, чем байт. Следующая таблица показывает согласование адресов Modbus для образа процесса в CPU.

Таблица 13-64 Согласование адресов Modbus для образа процесса

Функции Modbus						**S7-1200**	
Коды	Функция	Область данных	Диапазон адресов			Область данных	Адрес CPU
01	Читать биты	Вывод	1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7
02	Читать биты	Ввод	10001	до	18192	Образ процессов по вводу	От I0.0 до I1023.7
04	Чтение слов	Ввод	30001	до	30512	Образ процессов по вводу	От IW0 до IW1022
05	Записать бит	Вывод	1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7
15	Записать биты	Вывод	1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7

Коды функций входящих сообщений Modbus (3, 6, и 16) читают/записывают слова в регистре хранения Modbus, который может быть M памятью или блоком данных. Тип регистра хранения определяется параметром MB_HOLD_REG.

Примечание

Назначение параметра MB_HOLD_REG

Регистры хранения, определенные как arrays of word, integer, wide character, unsigned integer, byte, short integer, unsigned short integer, character, double word, double integer, unsigned double integer или real, могут находиться в любой области памяти.

Регистры хранения, определенные как структуры, должны находиться в не оптимизированных DB.

Для регистра хранения Modbus в M памяти следует использовать формат указателя Any. Это формат P# "Адрес бита" "Тип данных" "Длина". Пример: P#M1000.0 WORD 500.

Следующая таблица показывает примеры согласования адресов Modbus с регистрами хранения, используемые для кодов функций Modbus 03 (чтение слов), 06 (запись слов)

и 16 (запись слов). Фактический верхний предел адресов DB определен максимальным объемом рабочей памяти и объемом M памяти для каждой из моделей CPU.

Таблица 13-65 Примеры для согласования адресов Modbus с адресами в памяти CPU

Адрес Modbus	**Примеры для параметра MB_HOLD_REG**		
	P#M100.0 Word 5	**P#DB10.DBx0.0 Word 5**	**"Recipe".ingredient**
40001	MW100	DB10.DBW0	"Recipe".ingredient[1]
40002	MW102	DB10.DBW2	"Recipe".ingredient[2]
40003	MW104	DB10.DBW4	"Recipe".ingredient[3]
40004	MW106	DB10.DBW6	"Recipe".ingredient[4]
40005	MW108	DB10.DBW8	"Recipe".ingredient[5]

Заголовок прикладного протокола Modbus

Заголовок прикладного протокола Modbus включает в себя первые семь байт любого сообщения MODBUS TCP. Этот заголовок содержит идентификатор транзакции, идентификатор протокола, длину и идентификатор устройства. Ответное сообщение инструкции MB_SERVER содержит те же значения для идентификатора транзакции, идентификатора протокола и идентификатора устройства, которые были получены в сообщении запроса Modbus. Поле длины рассчитывается инструкцией MB_SERVER.

Подключение нескольких серверов

Может быть создано несколько подключений серверов. Одиночный PLC может установить параллельные соединения с несколькими клиентами Modbus TCP.

Сервер Modbus TCP может поддерживать параллельные соединения вплоть до максимального количества соединений для открытых коммуникационных соединений пользователя (Open user communication), разрешенных для устройства PLC. Общее количество соединений для устройства PLC, включая клиентов и сервер Modbus TCP, не должно превышать максимальное количество поддерживаемых открытых коммуникационных соединений пользователя (Страница 807). Соединения Modbus TCP могут совместно использоваться подключениям типа клиента и сервера.

Отдельные параллельные серверные соединения должны следовать данным правилам:

- Каждое соединение MB_SERVER должно использовать уникальный DB экземпляра.
- Каждое соединение MB_SERVER должно назначать уникальный номер IP-порта сервера. Для каждого порта поддерживается только одно соединение.
- Каждое соединение MB_SERVER должно назначать уникальный ID соединения.
- MB_SERVER должен вызываться индивидуально для каждого соединения (с соответствующим экземплярным DB).

ID соединения должен быть уникальным для каждого отдельного соединения. Для каждого отдельного DB экземпляра должен использоваться один ID соединения. DB

экземпляра и ID соединения составляют пару и должны быть уникальными для каждого соединения.

Таблица 13-66 Коды диагностических функций Modbus

Функции диагностики Modbus для MB_SERVER		
Коды	Подфункция	Описание
08	0x0000	Вывод данных запроса эхо-контроля: Инструкция MB_SERVER возвращает Modbus клиенту эхо полученного слова данных.
08	0x000A	Очистка счетчика событий коммуникации: Инструкция MB_SERVER очищает счетчик событий коммуникации, который используется для Modbus функции 11.
11		Вызов счетчика событий коммуникации: Инструкция MB_SERVER использует внутренний счетчик событий коммуникации для регистрации количества выполненных запросов чтения и записи Modbus, которые отправляются на сервер Modbus. Значение счетчика не увеличивается при выполнении функций 8, функции 11 и запросов, приведших к ошибке обмена данными. Функция широковещательной передачи недоступна для Modbus TCP, так как в любой момент имеется только одно соединение клиент-сервер.

Переменные в блоке данных (DB) инструкции MB_SERVER

В таблице ниже приведены общедоступные статические переменные в экземплярном блоке данных MB_SERVER, которые могут быть использованы в программе.

Таблица 13-67 Общедоступные статические переменные MB_SERVER

Переменная	**Тип данных**	**По умолчанию**	**Описание**
HR_Start_Offset	Word	0	Задает начальный адрес регистра хранения Modbus
Request_Count	Word	0	Количество всех полученных этим сервером запросов
Server_Message_Count	Word	0	Количество запросов, полученных для этого определенного сервера
Xmt_Rcv_Count	Word	0	Количество операций по передаче или приему данных, в которых произошли ошибки. Кроме того, увеличивается на единицу, если принято сообщение, которое является недопустимым сообщением Modbus.
Exception_Count	Word	0	Ошибки Modbus, требующие возвращенного исключения
Success_Count	Word	0	Количество запросов, полученных для этого определенного сервера без ошибок протокола.
Connected	Bool	0	Указывает, существует ли или нет соединение с назначенным клиентом. 1 = соединение установлено , 0 = соединение не установлено
QB_Start	UInt	0	Начальный адрес выходных байт, в которые может выполнять запись CPU (от AB0 до AB65535)
QB_Count	UInt	65535	Количество байт, в которые может выполнять запись удаленное устройство. Если QB_Count = 0, то удаленное устройство не может записывать на выходы. Пример: Если запись должна быть возможна только в AB10 - AB17, то установка QB_Start = 10 и QB_Count = 8.

Переменная	Тип данных	По умолчанию	Описание
QB_Read_Start	UInt	0	Начальный адрес выходных байт, из которых может выполнять чтение CPU (от AB0 до AB65535)
QB_Read_Count	UInt	65535	Количество выходных байт, из которых может считывать удаленное устройство. Если QB_Count = 0, то удаленное устройство не может читать из выходов. Пример: Если чтение должно быть возможным только для AB10 - AB17, то установка QB_Start = 10 и QB_Count = 8.
IB_Read_Start	UInt	0	Начальный адрес входных байт, из которых может выполнять чтение CPU (от EB0 до EB65535)
IB_Read_Count	UInt	65535	Количество входных байт, из которых может считывать удаленное устройство. Если IB_Count = 0, то удаленное устройство не может читать из входов. Пример: Если чтение должно быть возможным только для EB10 - EB17, то установка IB_Start = 10 и IB_Count = 8.
NDR_immediate	Bool	FALSE	Значение идентично параметру NDR (новые данные готовы). MB_SERVER обновляет "NDR_immediate" в том же вызове, в котором обрабатывается запрос на запись Modbus TCP.
DR_immediate	Bool	FALSE	Значение идентично параметру DR (чтение данных). MB_SERVER обновляет "DR_immediate" в том же вызове, в котором обрабатывается запрос на запись Modbus TCP.

Программа может записывать данные в операции управления сервера Modbus и следующие переменные:

- HR_Start_Offset
- QB_Start
- QB_Count
- QB_Read_Start
- QB_Read_Count
- QB_Read_Start
- QB_Read_Count

Для доступности переменных в блоке данных (DB) инструкции MB_SERVER существуют следующие требования к версиям

Таблица 13-68 Требования к версиям для доступности переменных в блоке данных (DB) инструкции MB_SERVER Инструкция, TIA Portal и S7-1200 CPU

Версия инструкции MB_SERVER	Версия TIA Portal	Версия прошивки S7-1200 CPU	Переменные блоков данных
4.2	V14 SP1	CPU FW V4.0 или выше	QB_Start
			QB_Count

Версия инструкции MB_SERVER	Версия TIA Portal	Версия прошивки S7-1200 CPU	Переменные блоков данных
5.0 или выше	V15 или выше	CPU FW V4.2 или выше	QB_Start
			QB_Count
			QB_Read_Start
			QB_Read_Count
			IB_Read_Start
			IB_Read_Count
			NDR_immediate
			DR_immediate

HR_Start_Offset

Адреса регистра хранения Modbus начинаются с 40001. Эти адреса соответствуют начальному адресу регистра хранения в памяти целевой системы. Однако можно использовать переменную HR_Start_Offset, чтобы сконфигурировать другой начальный адрес для регистра хранения Modbus, отличный от 40001.

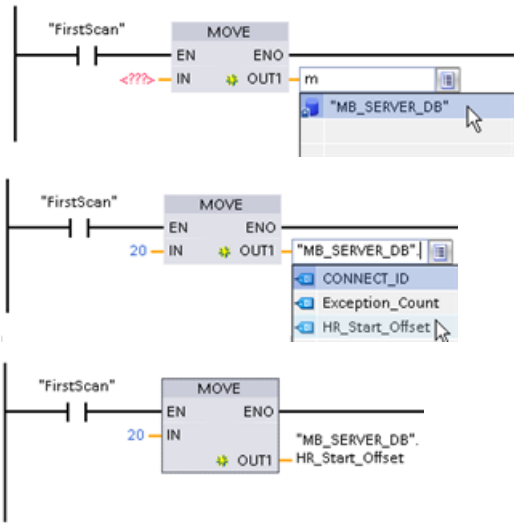
Можно, например, сконфигурировать регистр хранения с началом в MW100 и длиной в 100 слов. При помощи смещения 20 можно задать начальный адрес регистра хранения 40021 вместо 40001. Любой адрес меньше 40021 и больше 40119 приводит к ошибке адресации.

Таблица 13-69 Примеры для адресации регистра хранения Modbus:

HR_Start_Offset	Адрес	Минимум	Максимум
0	Адрес Modbus (Word)	40001	40099
	Адрес S7-1200	MW100	MW298
20	Адрес Modbus (Word)	40021	40119
	Адрес S7-1200	MW100	MW298

HR_Start_Offset - это данные Word в блоке данных экземпляра MB-SERVER, назначающие начальный адрес для регистра хранения Modbus. Эти общедоступные статические переменные можно выбрать из выпадающего списка параметров, после того как в программу была вставлена инструкция MB_SERVER.

Например, после вставки MB_SERVER в сегмент LAD, можно перейти в предыдущий сегмент и назначить HR_Start_Offset. Начальный адрес необходимо назначить до исполнения MB_SERVER.



Ввод переменной для сервера Modbus через имя DB по умолчанию:

1. Установить курсор в поле параметра и ввести символ m.
2. В выпадающем списке имен DB выбрать "MB_SERVER_DB".
3. В выпадающем списке переменных DB выбрать "MB_SERVER_DB.HR_Start_Offset".

Обращение к областям данных в блоках данных (DB) вместо прямого доступа к адресам Modbus

От версии V5.0 инструкции MB_SERVER и версии V4.2 прошивки S7-1200 CPU возможно обращение к областям данных в DB, вместо прямого обращения к образам процесса и регистрам хранения. Для этого на странице свойств "Атрибуты" глобального DB необходимо снять флажки "Помещать только в загружаемую память" и "Оптимизированный доступ к блоку".

Если поступает запрос Modbus и область данных для типа данных Modbus соответствующего кода функции не была определена, то инструкция MB_SERVER обрабатывает запрос согласно процедуре в прежних версиях инструкции: Можно обращаться напрямую к образам процесса и регистрам хранения.

Если область данных для типа данных Modbus кода функции была определена, то инструкция MB_SERVER выполняет чтение из этой области данных и выполняет запись в эту область данных. Процесс чтение или записи зависит от типа задания.

Примечание

Если область данных сконфигурирована, то инструкция MB_SERVER игнорирует смещения и области, сконфигурированные статическими переменными в блоке данных экземпляра, который соответствует типу данных в области данных. Эти смещения и области действительны только для образа процесса или памяти, на которые ссылается MB_HOLD_REG. Параметры начала и длины области данных предлагают собственный способ для определения смещений и областей.

Для индивидуального запроса Modbus возможна запись только в одну область данных или чтение из нее. Например, если требуется чтение регистров хранения, занимающих несколько областей данных, то потребуется несколько запросов Modbus.

Для определения областей данных действуют следующие правила:

- Можно определить до восьми областей данных в различных DB, при этом каждый DB может содержать только одну область данных. Для одного запроса Modbus возможна запись только в одну область данных или чтение из нее. Каждая область данных соответствует одному диапазону адресов MODBUS. Области данных устанавливаются в статической переменной "Data_Area_Array" экземплярного DB.
- Если необходимо использовать менее восьми областей данных, то следует разместить необходимые области данных одну за другой без пропусков. На первой пустой записи в областях данных поиск области данных во время обработки отменяется. Если, например, элементы поля 1, 2, 4 и 5 были определены как области данных, то "Data_Area_Array" распознает только элементы поля 1 и 2, поскольку элемент поля 3 пуст.

- После Data_Area_Array включает в себя восемь элементов: от Data_Area_Array[1] до Data_Area_Array[8]
- Каждый элемент поля Data_Area_Array[x], 1 <= x <= 8 - это UDT типа MB_DataArea со следующей структурой:

Параметр	Тип данных	Значение
data_ type	UInt	Идентификатор для типа данных MODBUS, который отображен на эту область данных: • 0: Идентификатор для пустого элемента поля или неиспользуемой области данных. В этом случае значения для db, start и length не имеют значения. • 1: Выход образа процесса (в комбинации с кодами функции 1, 5 и 15) • 2: Вход образа процесса (в комбинации с кодом функции 2) • 3: Регистр хранения (в комбинации с кодами функции 3, 6 и 16) • 4: Входной регистр (в комбинации с кодом функции 4) Примечание: Если для типа данных MODBUS была определена область данных, то инструкция MB_SERVER больше не сможет напрямую обращаться к этому типу данных MODBUS. Если адрес запроса MODBUS для такого типа данных не соответствует установленной области данных, то в STATUS возвращается значение W#16#8383.
db	UInt	Номер блока данных, на который отображаются регистр MODBUS или определенные после биты Номер DB должен быть оригинальным в областях данных. Номер DB не может встречаться в нескольких областях данных. На странице свойств "Атрибуты" глобального DB необходимо снять флажки "Помещать только в загружаемую память" и "Оптимизированный доступ к блоку". Области данных также начинаются с адрес байта 0 в DB. Допустимые значения: от 1 до 60999
start	UInt	Первый адрес MODBUS, отображенный на блок данных, начинающийся с адреса 0.0 Допустимые значения: от 0 до 65535
length	UInt	Число бит (для значений 1 и 2 из data_type) или число регистров (для значений 3 и 4 из data_type) Диапазоны адресов MODBUS одного типа данных MODBUS не должны накладываться друг на друга. Допустимые значения: От 1 до 65535

Примеры для определения областей данных:

- Первый пример: data_type = 3, db = 1, start = 10, length = 6
  CPU отображает регистры хранения (data_type = 3) в блоке данных 1 (db = 1) и устанавливает Modbus адрес 10 (start = 10) на слово данных 0, а последний действительный Modbus адрес 15 (length = 6) на слово данных 5.
- Второй пример: data_type = 2, db = 15, start = 1700, length = 112
  CPU отображает входы (data_type = 2) в блоке данных 15 (db = 15) и устанавливает Modbus адрес 1700 (start = 1700) на слово данных 0, а последний действительный Modbus адрес 1811 (length = 112) на слово данных 111.

Коды условий

Таблица 13-70 Коды состояния выполнения MB_SERVER [1]

STATUS (W#16#....)	Код ответа серверу Modbus (B#16#)	Ошибка протокола Modbus
7001		MB_SERVER ожидает, что клиент Modbus установит соединение с назначенным TCP портом. Этот код возвращается для первого выполнения операции подключения или отключения.
7002		MB_SERVER ожидает, что клиент Modbus установит соединение с назначенным TCP портом. Этот код будет возвращен для любого последующего выполнения при ожидании завершения операции подключения или отключения.
7003		Операция отключения успешно завершена (действует только для одного цикла PLC)
8187		MB_HOLD_REG является недействительным, возможно указывает на оптимизированный DB или на область меньше 2 байт.
818C		Указатель MB_HOLD_REG ссылает на оптимизированною область DB (при этом речь должна идти об не оптимизированной области DB или области M памяти), или тайм-аут для заблокированного процесса превышает предельное значение в 55 секунд (спец. для S7-1200)
8381	01	Код функции не поддерживается
8382	03	Ошибка в длине данных
8383	02	Ошибка в адресе данных или доступ вне диапазона адресов MB_HOLD_REG
8384	03	Ошибка в значении данных

STATUS (W#16#....)	Код ответа серверу Modbus (B#16#)	Ошибка протокола Modbus
8385	03	Диагностический код не поддерживается (код функции 08)
8389		Недействительное определение для области данных: • Недопустимое значение для типа данных • Номер DB недействителен или отсутствует: – Недействительное значение DB – Номер DB отсутствует – Номер DB уже использует в другой области данных – DB с оптимизированным доступом – DB находится не в рабочей памяти • Недействительное значение длины • Наложение диапазонов адресов MODBUS, относящихся к одному типу данных MODBUS

[1] Наряду с упомянутыми выше ошибкам MB_SERVER, ошибки могут сигнализироваться и из базовых коммуникационных инструкций T блока (TCON, TDISCON, TSEND и TRCV).

MB_RED_CLIENT (резервируемая коммуникация через PROFINET в качестве клиента Modbus TCP)

Можно использовать эту инструкцию для создания соединения между S7-1200 CPU и устройством с поддержкой протокола MODBUS TCP.

Таблица 13-71 Инструкция MB_RED_CLIENT

LAD/FBD	SCL	Описание

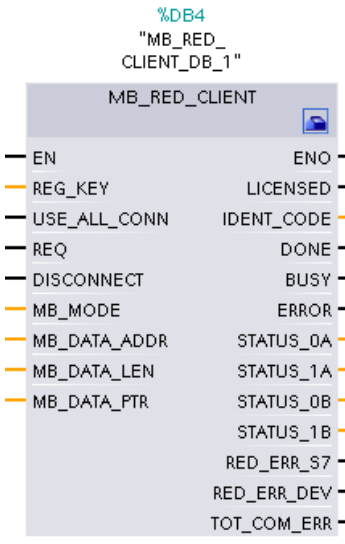
	`"MB_RED_CLIENT_DB"(`
	`REG_KEY:=_string_in_,`
	`USE_ALL_CONN:=_bool_in_`
	`REQ:=_bool_in_,`
	`DISCONNECT:=_bool_in_,`
	`MB_MODE:=_usint_in_,`
	`MB_DATA_ADDR:=_udint_in_,`
	`MB_DATA_LEN:=_uint_in_,`
	`LICENSED=>_bool_out_`
	`IDENT_CODE=>_string_out_`
	`DONE=>_bool_out_,`
	`BUSY=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS_0A=>_word_out_,`
	`STATUS_1A=>_word_out_,`
	`STATUS_0B=>_word_out_,`
	`STATUS_1B=>_word_out_,`
	`RED_ERR_S7=>_bool_out_,`
	`RED_ERR_DEV=>_bool_out_,`
	`TOT_COM_ERR=>_bool_out_,`
	`MB_DATA_PTR:=_variant_inout_);`
		Инструкция MB_RED_CLIENT выполняет обмен данными в качестве клиента Modbus TCP через соединение PROFINET.
		С помощью инструкции MB_RED_CLIENT можно установить резервируемое соединение между клиентом и сервером, отправлять запросы Modbus, получать ответы и управлять завершением соединения через клиента Modbus TCP.

Таблица 13-72 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REG_KEY [1]	IN	STRING[17]	Регистрационный код для лицензирования Инструкция MB_RED_CLIENT должна лицензироваться индивидуально на каждом CPU.
USE_ALL_CONN	IN	Bool	Указать число сконфигурированных соединений, через которые должна передаваться телеграмма: • 0: Всегда передавать телеграмму по одному единственному соединению, переключаться на следующее соединение только при ошибке • 1: Передавать телеграмму по всем сконфигурированным соединениям
REQ	IN	Bool	Запрос Modbus на сервер Modbus TCP Параметр REQ управляется по уровню. Т.е. пока вход установлен (REQ = TRUE), инструкция отправляет запросы связи. Если соединение еще не установлено, оно устанавливается сейчас, и телеграмма Modbus отправляется сразу после этого. Изменения входных параметров не вступят в силу до тех пор, пока сервер не ответит или не будет возвращено сообщение об ошибке. Если параметр REQ устанавливается снова во время текущего запроса Modbus, дополнительная передача не выполняется.
DISCONNECT	IN	Bool	Это параметр управляет созданием и завершением соединения с сервером Modbus: • 0: Создание коммуникационного соединения с партнером, сконфигурированным в параметре CONNECT (см. параметр CONNECT). • 1: Разрыв коммуникационного соединения. Во время разрыва соединения не выполняются никакие другие функции. После успешного разрыва соединения в параметре STATUS_x возвращается значение 0003.
MB_MODE [2]	IN	USInt	Выбирает режим запроса Modbus (считывание, запись или диагностика) или прямой выбор функции Modbus
MB_DATA_ADDR [2]	IN	UDInt	Адрес Modbus в зависимости от MB_MODE
MB_DATA_LEN	IN	UInt	Длина данных: количество бит или регистров для доступа к данным
MB_DATA_PTR [2]	IN_OUT	Variant	Указатель на буфер данных для принятых от сервера Modbus данных или передаваемых на сервер Modbus данных.
LICENSED [1]	OUT	Bool	• 0: Инструкция не лицензирована • 1: Инструкция лицензирована
IDENT_CODE [1]	OUT	STRING[18]	Идентификация для лицензирования. Использовать следующую строку для запроса регистрационного кода REG_KEY.
DONE	OUT	Bool	Бит в выходном параметре DONE устанавливается на "1", если активное задание Modbus было выполнено без ошибок как минимум на одном соединении.
BUSY	OUT	Bool	• 0: Нет обрабатываемых запросов Modbus • 1: Запрос Modbus обрабатывается Выходной параметр BUSY не установлен при создании и разрыве соединения.

Параметр и тип данных		Тип данных	Описание
ERROR	OUT	Bool	• 0: Ошибки отсутствуют • 1: Не удалось передать активную операцию Modbus ни на одно из сконфигурированных соединений. Причина отображается в параметре STATUS_x.
STATUS_0A [3]	OUT	Word	Подробная информация о состоянии инструкции по соединению 0A.
STATUS_1A [3]	OUT	Word	Подробная информация о состоянии инструкции по соединению 1A.
STATUS_0B [3]	OUT	Word	Подробная информация о состоянии инструкции по соединению 0B.
STATUS_1B [3]	OUT	Word	Подробная информация о состоянии инструкции по соединению 1B.
RED_ERR_S7 [3]	OUT	Bool	• 0: Ошибки резервирования в SIMATIC отсутствуют • 1: Ошибка резервирования в SIMATIC
RED_ERR_S7 [3]	OUT	Bool	• 0: Ошибки резервирования на стороне партнёра по соединению отсутствуют • 1: Ошибка резервирования на стороне партнёра по соединению
RED_ERR_S7 [3]	OUT	Bool	• 0: Как минимум 1 сконфигурированное соединение установлено • 1: Полная потеря связи, все сконфигурированные соединения разорваны

[1] Более подробная информация содержится в следующем разделе "Лицензирование".

[2] Более подробная информация содержится в следующем разделе "Входные параметры:MB_MODE, MB_DATA_ADDR, MB_DATA_LEN и MB_DATA_PTR".

[3] Более подробная информация содержится в следующем разделе "Выходные параметры:STATUS_x, RED_ERR_S7, RED_ERR_DEV и TOT_COM_ERR".

Примечание

Согласованные входные данные при вызове MB_RED_CLIENT

При вызове клиентской инструкции Modbus, значения входных параметров сохраняются в системе. При обработке телеграммы значения не должны изменяться.

Примечание

Требования к версии прошивки CPU

Инструкции Modbus TCP, описанные в этом разделе руководства, требуют прошивки версии V4.2 или выше.

Для использования инструкции дополнительный аппаратный модуль не требуется.

Подключение нескольких клиентов

CPU могут обрабатывать клиентских соединений Modbus TCP. Макс. число соединений зависит от используемого CPU и указано в его технических параметрах. Общее количество соединений CPU, включая клиентов и серверы Modbus TCP, не должно превышать максимальное количество поддерживаемых соединений.

Для отдельных клиентских соединений следует придерживаться следующих правил:

- Каждое соединение "MB_RED_CLIENT " должно использовать уникальный DB экземпляра.
- Для каждого соединения MB_RED_CLIENT должен быть указан уникальный IP-адрес сервера.
- Каждому соединению "MB_RED_CLIENT " требуется уникальный ID соединения. Соединение должен быть уникальным для всего CPU.

Эксплуатация и резервирование

Участники обмена данными могут быть выполнены как автономные или резервируемые узлы. Если один из партнеров является автономным, то речь идет об одностороннем резервировании. Если оба партнера являются резервируемыми, то речь идет о двустороннем резервировании:

- Одностороннее резервирование:
  - Описание: Для каждого соединения между партнерами должна быть сконфигурирована отдельная линия. Точки подсоединения **SIMATIC S7** обозначаются как **0** и **1**. Точки подсоединения **партнеров** обозначаются как **A** и **B**. R-CPU или H-CPU 1 относится к точке подсоединения 0, R-CPU или H-CPU 2 относится к точке подсоединения 1.
  - Конфигурация: В резервируемой конфигурации S7 создается соединение S7 точки подсоединения 0 с узловой точкой A партнёра по соединению (подключение S7 точки подсоединения **0** к партнеру/узлу **A** => соединение **0A**), и создается соединение S7 точки подсоединения 1 с узловой точкой A партнёра по соединению (подключение S7 точки подсоединения **1** к партнеру/узлу **A** => соединение **1A**). На рисунке показаны обозначения соединения.

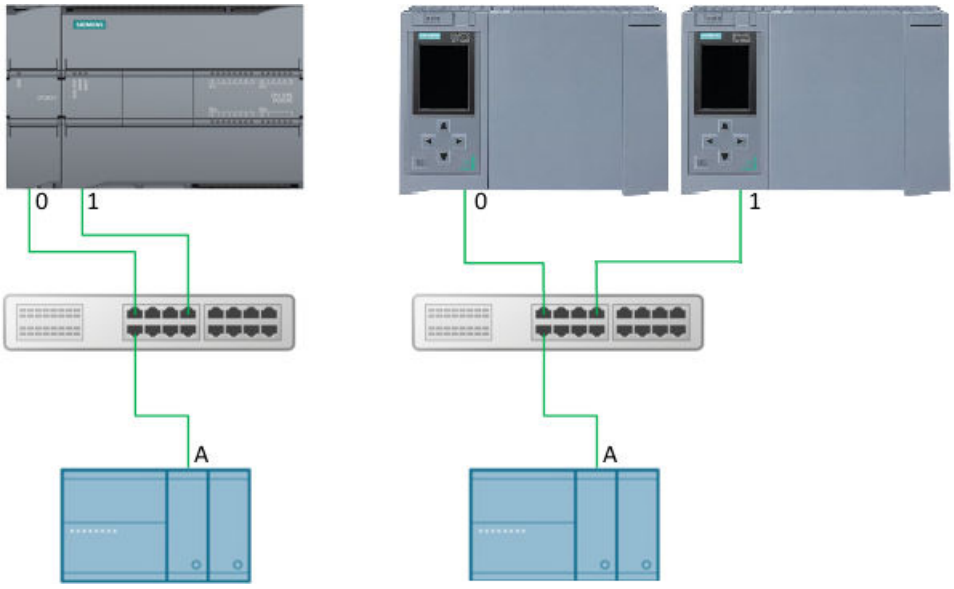


Рисунок 13-1 Одностороннее резервирование S7

– Если S7 является автономным, а партнёр по соединению резервируемым, то создается соединение S7 точки подсоединения 0 с узловой точкой A партнёра по соединению (подключение S7 точки подсоединения **0** к партнеру/узлу **A** => соединение **0A**), и создается соединение S7 точки подсоединения 0 с узловой точкой B партнёра по соединению (подключение S7 точки подсоединения **0** к партнеру/узлу **B** => соединение **0B**). На рисунке показаны обозначения соединения.

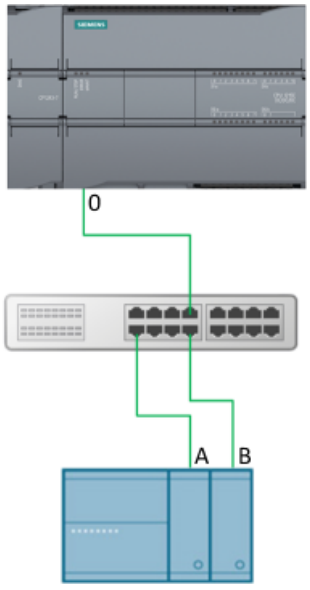


Рисунок 13-2 Одностороннее резервирование, партнер

- Двухстороннее резервирование:
  – Описание: Для каждого соединения между партнерами должна быть сконфигурирована отдельная линия. Точки подсоединения **SIMATIC S7** обозначаются как 0 и **1**. Точки подсоединения **партнеров** обозначаются как **A** и **B**. R-CPU или H-CPU 1 относится к точке подсоединения 0, R-CPU или H-CPU 2 относится к точке подсоединения 1.

– Конфигурация: При двухстороннем резервировании создается два подключения с началом в точке подсоединения 0 (подключение S7 точки подсоединения **0** к парнеру/узлу **A** => соединение **0A** и подключение S7 точки подсоединения**0** к парнеру/узлу **B** => соединение **0B**), а также создается два подключения с началом в S7 точке подсоединения 1 к узловым точкам A и B партнёра по соединению (подключение S7 точки подсоединения **1** к парнеру/узлу **A** => соединение **1A** и подключение S7 точки подсоединения **1** к парнеру/узлу **B** => соединение **1B**). На рисунке показаны обозначения соединения.

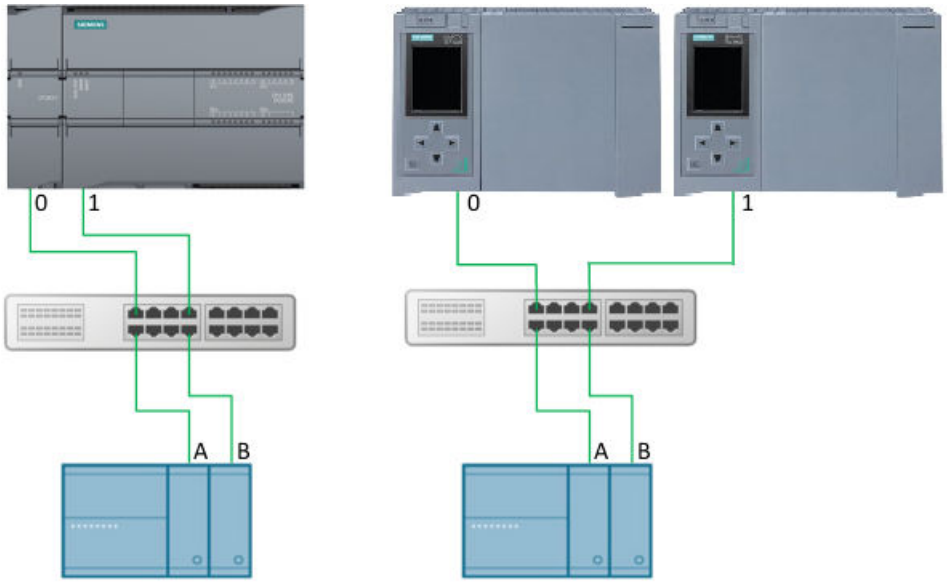


Рисунок 13-3 Двухстороннее резервирование:

- Обработка телеграмм: Телеграммы могут передавать по одному или по всем сконфигурированным соединениям:
  - Передача телеграмм по одному соединению: Телеграмма MODBUS с установкой USE_ALL_CONN = FALSE передается по одному - активному в настоящий момент - соединению. При превышении времени (нет ответа от сервера) или при ошибке соединения предпринимается попытка отправить телеграмму по другим (максимум 4) сконфигурированным соединениям. Последовательность в этом случае 0A, 1A, 0B и 1B. Если телеграмма была успешно передана через одно из соединений, это соединение помечается как "активное", и дальнейший обмен телеграммами выполняется через это соединение. В случае ошибки активного соединения снова предпринимается попытка отправить телеграмму через все настроенные соединения. Если все попытки отправки терпят неудачу, то ERROR и STATUS_x устанавливаются соответственно.
    После получения ответной телеграммы выполняется проверка достоверности. Если эта проверка пройдена успешно, то необходимые действия выполняются, и задание считается выполненным без ошибок. Выход DONE устанавливается. Если во время проверки обнаружены ошибки, то задание завершается с ошибками, устанавливается бит ERROR, и номер ошибки отображается в STATUS_x. В этом случае больше не предпринимается попыток отправить телеграмму по следующему настроенному соединению. Переключение на другие настроенные соединения выполняется только в том случае, если обнаружена ошибка соединения или не получен ответ.
  - Передача телеграмм по всем соединениям: Телеграмма MODBUS с установкой USE_ALL_CONN = TRUE передается по всем сконфигурированным, установленным соединениям. Проверка достоверности выполняется после того, как ответная телеграмма была получена по одному из соединений. Если эта проверка успешна, выполняются необходимые действия. Если правильная ответная телеграмма была получена хотя бы на одном соединении, то устанавливается выход DONE.
- Выходы резервирования RED_ERR_S7, RED_ERR_DEV и TOT_COM_ERR:

– Биты резервирования RED_ERR_S7, RED_ERR_DEV и TOT_COM_ERR устанавливаются в зависимости от статуса выходов состояния:

Number of faulty connections	STATUS_0A	STATUS_0B	STATUS_1A	STATUS_1B	RED_ERR_S7	RED_ERR_DEV	TOT_COM_ERR
0	okay	okay	okay	okay	FALSE	FALSE	FALSE
1	okay	okay	okay	Error	FALSE	FALSE	FALSE
	okay	okay	Error	okay	FALSE	FALSE	FALSE
	okay	Error	okay	okay	FALSE	FALSE	FALSE
	Error	okay	okay	okay	FALSE	FALSE	FALSE
2	okay	okay	Error	Error	TRUE	FALSE	FALSE
	okay	Error	okay	Error	FALSE	TRUE	FALSE
	Error	okay	okay	Error	FALSE	FALSE	FALSE
	okay	Error	Error	okay	FALSE	FALSE	FALSE
	Error	okay	Error	okay	FALSE	TRUE	FALSE
	Error	Error	okay	okay	TRUE	FALSE	FALSE
3	Error	Error	Error	okay	TRUE	TRUE	FALSE
	Error	Error	okay	Error	TRUE	TRUE	FALSE
	Error	okay	Error	Error	TRUE	TRUE	FALSE
	okay	Error	Error	Error	TRUE	TRUE	FALSE
4	Error	Error	Error	Error	TRUE	TRUE	TRUE

Рисунок 13-4 Отображение битов прерывания для установки резервирования с обеих сторон

Number of faulty connections	STATUS_0A	STATUS_0B	STATUS_1A	STATUS_1B	RED_ERR_S7	RED_ERR_DEV	TOT_COM_ERR
0	okay	0AFF	okay	0AFF	FALSE	FALSE	FALSE
1	okay	0AFF	Error	0AFF	TRUE	TRUE	FALSE
	Error	0AFF	okay	0AFF	TRUE	TRUE	FALSE
2	Error	0AFF	Error	0AFF	TRUE	TRUE	TRUE

Рисунок 13-5 Отображение битов прерывания для установки резервирования с одной сторон

Примечание

Номера портов для клиента и сервера

Modbus клиент используется номер порта, начинающий на 2000. Для обращения к Modbus серверу обычно используется порт с номером 502.

Параметрирование

Для S7-1200 можно использовать инструкцию MB_RED_CLIENT **V1.0** и **V1.1**. CPU реализует соединения через локальный интерфейс CPU или CM/CP. Через структуру TCON_IP_V4 соединения конфигурируются CPU и устанавливаются.

Конфигурация MB_RED_CLIENT: Следующие установки выполняются в диалоговом окне конфигурирования для инструкции MB_RED_CLIENT:

- Параметры для соединений 0A, 1A, 0B и 1B (дополнительную информацию о конфигурации резервирования можно найти в разделе "Эксплуатация и резервирование" выше)
- Внутренние параметры (необязательно)

Для открытия диалогового окна конфигурирования можно использовать инструкцию MB_RED_CLIENT или технологические объекты:

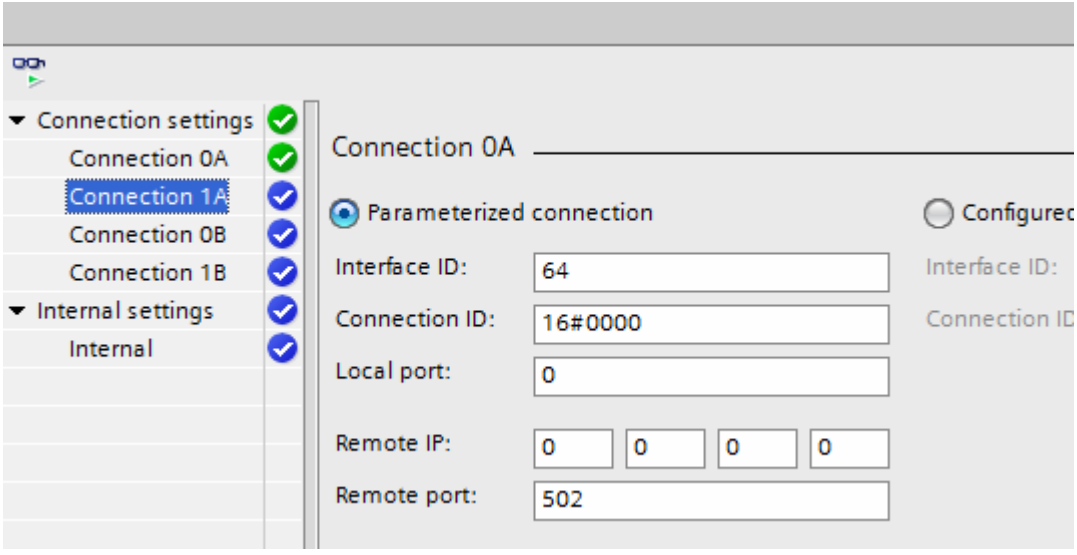


Рисунок 13-6 Настроенное клиентское соединение

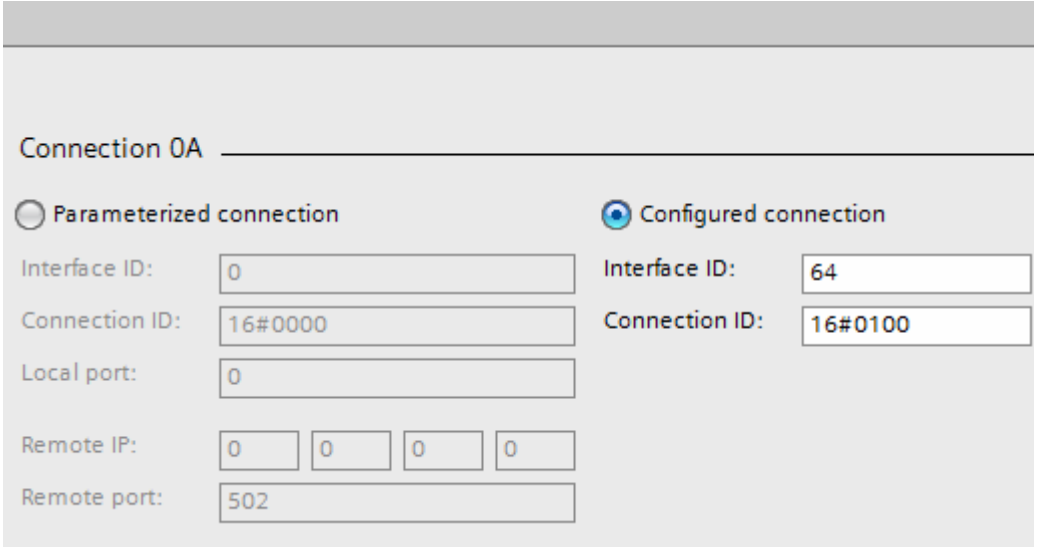


Рисунок 13-7 Сконфигурированное клиентское соединение

Переменная (тег)	Стартовое значение	Описание
Сконфигурированные соединения		
Interface ID	64	Аппаратный идентификатор используемого PN-интерфейса
Connection ID	16#0000	ID соединений для используемых соединений Эти ID соединений не должны повторяться в рамках CPU.
Local port	0	Локальный номер порта клиента. По умолчанию номер порта для клиента не введен.
Remote IP	0.0.0.0	Удаленный IP-адрес сервера
Remote port	502	Удаленный номер порта сервера Порт по умолчанию для Modbus/TCP сервера 502.
Сконфигурированные соединения		
Interface ID	64	Аппаратный идентификатор используемого PN-интерфейса
Connection ID	16#0000	ID соединений для используемых соединений Эти соединения конфигурируются в обзоре сетей.

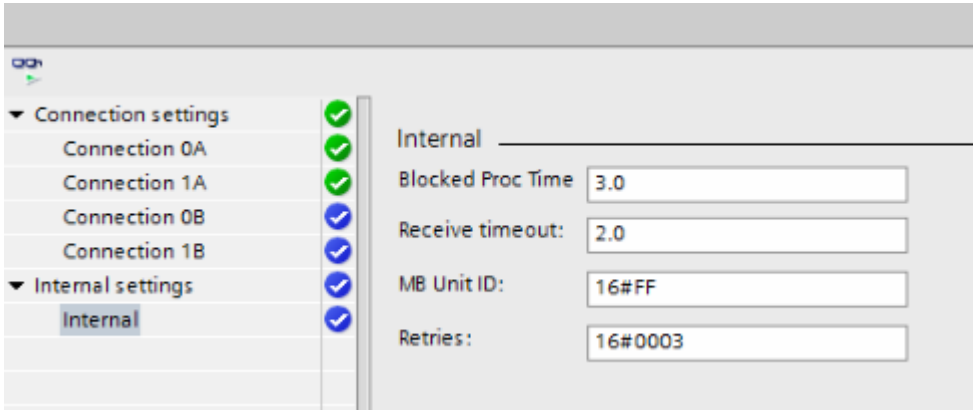


Рисунок 13-8 Внутренние параметры (необязательно)

Переменная (тег)	Тип данных	Стартовое значение	Описание
Blocked Proc Time	REAL	3.0	Время ожидания в секундах, прежде чем при блокируемом экземпляре Modbus будет сброшена статическая переменная ACTIVE. Это может, например, произойти, если выводится клиентский запрос и выполнение клиентской функции прерывается до того, как запрос был полностью выполнен. Время ожидания должно составлять от 0,5 до 55 сек.
Receive timeout	REAL	2.0	Интервал времени в секундах, в течение которого инструкция "MB_RED_CLIENT" ожидает ответа сервера. Интервал должен составлять от 0,5 до 55 сек.

Переменная (тег)	Тип данных	Стартовое значение	Описание
MB_Unit_ID	BYTE	255	Идентификатор оборудования Modbus: Сервер Modbus TCP адресуется с помощью своего IP-адреса. Поэтому параметр MB_UNIT_ID не используется при адресации Modbus TCP. Параметр MB_UNIT_ID соответствует полю адреса ведомого устройства в протоколе Modbus RTU. Если сервер Modbus/TCP сервер используется в качестве шлюза к протоколу Modbus RTU, MB_UNIT_ID может использоваться для идентификации ведомого устройства, подключенного к последовательной сети. Параметр MB_UNIT_ID в этом случае передает запрос по корректному адресу ведомого устройства Modbus RTU. Следует помнить, что некоторым Modbus/TCP устройствам для инициализации в рамках ограниченного диапазона значений может потребоваться параметр MB_UNIT_ID.
Retries	WORD	3	Число попыток передачи инструкции MB_RED_CLIENT до возврата ошибки W#16#80C8.

Примечание

Переменная (тег) MB_Transaction_ID

Если ID транзакции в ответе сервере Modbus TCP не совпадает с ID транзакции в задании MB_RED_CLIENT, то инструкция MB_RED_CLIENT запускает время ожидания RCV_TIMEOUT * RETRIES для ответа от сервера Modbus TCP с правильным ID транзакции. Как только это время истекает, возвращается ошибка W#16#80C8.

Лицензирование

Инструкция MB_RED_CLIENT является платной и подлежит лицензированию для каждого CPU по отдельности. Лицензирование выполняется за два этапа:

- Отображение IDENT_CODE лицензии
- Ввод регистрационного ключа REG_KEY: Необходимо ввести регистрационный ключ REG_KEY для каждой инструкции MB_RED_CLIENT. Сохранить REG_KEY в глобальный блок данных, из которого все инструкции MB_RED_CLIENT получают требуемый регистрационный ключ.

Порядок действий для отображения IDENT_CODE лицензии:

1. Задать параметры инструкции MB_RED_CLIENT в соответствии с требованиями в циклическом OB. Загрузить программу в CPU и перевести CPU в RUN.
2. Открыть DB экземпляра Modbus инструкции и нажать кнопку "Показать все".

3. Экземплярный блок данных показывает на выходе IDENT_CODE 18-значную последовательность символов.

	Name	Data type	Start value	Monitor value
1	Input			
2	REG_KEY	String[17]	''	''
3	USE_ALL_CONN	Bool	false	FALSE
4	REQ	Bool	false	FALSE
5	DISCONNECT	Bool	false	FALSE
6	MB_MODE	USInt	0	0
7	MB_DATA_ADDR	UDInt	0	0
8	MB_DATA_LEN	UInt	0	0
9	Output			
10	LICENSED	Bool	false	FALSE
11	IDENT_CODE	String[18]	''	'RTPCFIGDCDIIHJHAH4'
12	DONE	Bool	false	FALSE
13	BUSY	Bool	false	FALSE
14	ERROR	Bool	false	FALSE

Рисунок 13-9 Лицензия

4. Скопировать эту последовательность символов с помощью функции Копировать/вставить из блока данных и вставить ее в форму (которая была получена после заказа продукта по электронной почте или находилась на CD).
5. Отправить форму через запрос на обслуживание на Клиентская поддержка (https://support.industry.siemens.com/my/ww/en/requests/#createRequest). В ответе придет регистрационный ключ для CPU.

Порядок действий для ввода регистрационного ключа REG_KEY:

1. Вставить через "Добавить новый блок..." новый глобальный блок данных с однозначным символьным именем, напр., "Лицензионный DB".
2. Создать в этом блоке параметр REG_KEY с типом данных STRING[17].

LICENSE_DB

	Name	Data type	Offset	Start value
1	Static			
2	REG_KEY	String[17]	0.0	''

Рисунок 13-10 REG KEY

3. Скопировать полученный 17-значный регистрационный ключ с помощью функции Копировать/вставить в столбец "Стартовое значение".
4. Ввести в циклическом OB в параметре REG_KEY инструкции MB_RED_CLIENT имя лицензионного DB и имя последовательности символов (напр., License_DB.REG_KEY).
5. Загрузить измененные блоки в CPU. Регистрационный ключ может быть введен при работе. Перевод CPU из STOP в RUN не требуется.
6. Теперь Modbus/TCP коммуникация через инструкцию MB_RED_CLIENT лицензирована для этого CPU, выходной бит LICENSED теперь TRUE.

Порядок действий для исправления отсутствующего или неправильного лицензирования:

- Если ввести неправильный регистрационный ключ или не вводить регистрационный ключ, то начинает мигать ERROR LED на CPU. Кроме этого, S7-1200 CPU выполняет циклическую запись в буфер диагностики с информацией об отсутствии лицензии.

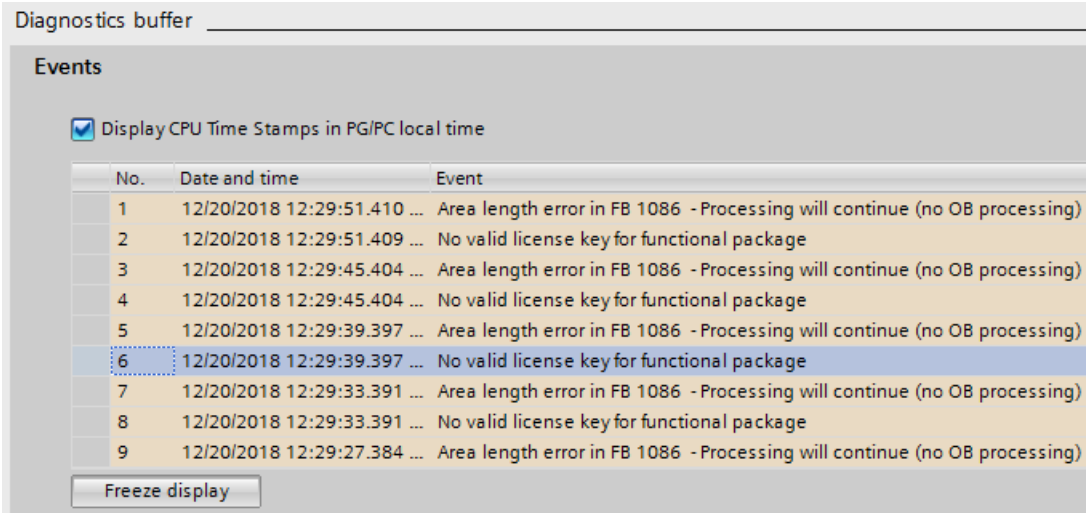


Рисунок 13-11 Буфер диагностики

- В случае отсутствующего или неправильного регистрационного ключа CPU продолжает обрабатывать коммуникацию Modbus TCP. Но на выходе STATUS_x непрерывно отображается "W#16#0A90" (нет действующего лицензионного ключа для пакета функций). Выходной бит LICENSED = FALSE.

Входные параметры: MB_MODE, MB_DATA_ADDR, MB_DATA_LEN и MB_DATA_PTR

Комбинация из MB_MODE, MB_DATA_ADDR и MB_DATA_LEN определяет код функции, используемый в текущем сообщении Modbus:

- **MB_MODE** содержит информацию, должно ли выполняться чтение или запись:
  Чтение: MB_MODE = 0, 101, 102, 103 и 104
  Запись: MB_MODE = 1, 2, 105, 106, 115 и 116 (Примечание: При MB_MODE = 2, нет различия между Modbus функциями 15 и 05 или между Modbus функциями 16 и 06.)
- **MB_DATA_ADDR** содержит информацию, какая информация должна быть считана или записана, а также информацию об адресах, на основе которой инструкция MB_RED_CLIENT вычисляет удаленный адрес.
- Параметр **MB_DATA_LEN** содержит количество считываемых/записываемых символов.

В следующей таблице представлены отношения между входными параметрами MB_MODE, MB_DATA_ADDR, MB_DATA_LEN инструкции MB_RED_CLIENT и функцией Modbus:

MB_MODE	MB_DATA_ADDR	MB_DATA_LEN	Функция Modbus	Функция и тип данных
0	От 1 до 9999	От 1 до. 2000	01	Чтение от 1 до 2000 выходных бит по удаленному адресу 0 - 9998
0	От 10001 до 19999	От 1 до 2.000	02	Чтение от 1 до 2.000 входных бит по удаленному адресу 0 - 9.998
0	• От 40.001 до 49.999 • От 400.001 до 465.535	От 1 до 125	03	• Чтение от 1 до 125 регистров хранения по удаленному адресу 0 - 9.998 • Чтение от 1 до 125 регистров хранения по удаленному адресу 0 - 65.534
0	От 30.001 до 39.999	От 1 до 125	04	Чтение от 1 до 125 входных слов по удаленному адресу 0 - 9.998
1	От 1 до 9.999	1	05	Запись 1 выходного бита по удаленному адресу 0 - 9.998
1	• От 40.001 до 49.999 • От 400.001 до 465.535	1	06	• Запись 1 регистра хранения по удаленному адресу 0 - 9.998 • Запись 1 регистра хранения по удаленному адресу 0 - 65.534
1	От 1 до 9.999	От 2 до 1.968	15	Запись от 2 до 1.968 выходных бит по удаленному адресу 0 - 9.998
1	• От 40.001 до 49.999 • От 400.001 до 465.535	От 2 до 123	16	• Запись от 2 до 123 регистров хранения по удаленному адресу 0 - 9.998 • Запись от 2 до 123 регистров хранения по удаленному адресу 0 - 65.534
2	От 1 до 9.999	От 1 до 1.968	15	Запись от 1 до 1.968 выходных бит по удаленному адресу 0 - 9.998
2	• От 40.001 до 49.999 • От 400.001 до 465.535	От 1 до 123	16	• Запись от 1 до 123 регистров хранения по удаленному адресу 0 - 9.998 • Запись от 1 до 123 регистров хранения по удаленному адресу 0 - 65.534
11	Инструкция не анализирует параметры MB_DATA_ADDR и MB_DATA_LEN, если эта функция выполняется.		11	Чтение слова состояния и счетчика событий сервера: • Слово состояния представляет статус обработки (0 - обработка не выполняется, 0xFFFF - обработка выполняется). • Счетчик событий увеличивается на единицу после успешного выполнения запроса Modbus. Если при выполнении функция Modbus возникла ошибка, то сервер отправляет сообщение, но счетчик событий не увеличивает свое значение.

MB_MODE	MB_DATA_ADDR	MB_DATA_LEN	Функция Modbus	Функция и тип данных
80	-	1	08	Проверка статуса сервера с помощью диагностического кода 0x0000 (петлевой тест - сервер возвращает запрос): 1 WORD на вызов
81	-	1	08	Сброс счетчика событий с помощью диагностического кода 0x000A: 1 WORD на вызов
101	От 0 до 65.535	От 1 до 2.000	01	Чтение от 1 до 2.000 выходных бит по удаленному адресу 0 - 65.535
102	От 0 до 65.535	От 1 до 2.000	02	Чтение от 1 до 2.000 входных бит по удаленному адресу 0 - 65.535
103	От 0 до 65.535	От 1 до 125	03	Чтение от 1 до 125 регистров хранения по удаленному адресу 0 - 65.535
104	От 0 до 65.535	От 1 до 125	04	Чтение от 1 до 125 входных слов по удаленному адресу 0 - 65.535
105	От 0 до 65.535	1	05	Запись 1 выходного бита по удаленному адресу 0 - 65.535
106	От 0 до 65.535	1	06	Запись 1 регистра хранения по удаленному адресу 0 - 65.535
115	От 0 до 65.535	От 1 до 1.968	15	Запись от 1 до 1.968 выходных бит по удаленному адресу 0 - 65.535
116	От 0 до 65.535	От 1 до 123	16	Запись от 1 до 123 регистров хранения по удаленному адресу 0 - 65.535
3 до 10, 12 до 79, 82 до 100, 107 до 114, 117 до 255				Зарезервировано

Пример:

Переменная (тег)	Значение
MB_MODE = 1 MB_DATA_ADDR = 1 MB_DATA_LEN = 1	Запись 1 выходного бита с кодом функции 5, начиная с удаленного адреса 0.
MB_MODE = 1 MB_DATA_ADDR = 1 MB_DATA_LEN = 2	Запись 2 выходных бит с кодом функции 15, начиная с удаленного адреса 0.
MB_MODE = 104 MB_DATA_ADDR = 17834 MB_DATA_LEN = 125	Запись 125 входных слов с кодом функции 4, начиная с удаленного адреса 17.834.

MB_DATA_PTR:

Параметр MB_DATA_PTR - это указатель на буфер данных для принимаемых от сервера Modbus данных или передаваемых на сервер Modbus данных. Можно использовать глобальный блок данных или область M памяти как буфер данных.

Следует использовать для буфера в области M памяти указатель в формате ANY следующим образом: "P#Bitadresse" "Тип данных" "Длина" (пример: P#M1000.0 WORD 500)

В зависимости от области памяти, в которой находится буфер данных, MB_DATA_PTR может ссылаться на различные структуры данных:

- Если используется глобальный DB с оптимизированным доступом, то MB_DATA_PTR может ссылаться на переменную (тег) с элементарным типом данных или на массив из элементарных типов данных. Поддерживаются следующие типы данных:

Тип данных	Длина в битах
Bool	1
Byte, SInt, USInt, Char	8
Word, Int, WChar, UInt	16
DWord, DInt, UDInt, Real	32

  Можно использовать все поддерживаемые типы данных для всех функций Modbus. Пример: MB_RED_CLIENT также может записывать полученный бит в переменной (теге) типа Byte на указанный адрес без изменения других битов в этом байте. Следовательно, необязательно иметь массив битов для выполнения бит-ориентированных функций.

- Если область адресов битовой памяти или глобальный DB со стандартным доступом используются в качестве области памяти, то это снимает все ограничения на элементарные типы данных для MB_DATA_PTR; в этом случае MB_DATA_PTR может ссылаться и на сложные структуры данных, такие как типы данных PLC (UDT) и системные типы данных (SDT).

Примечание

Использование области адресов битовой памяти в качестве буфера данных

Если для MB_DATA_PTR используется область адресов битовой памяти в качестве буфера данных, то следует учитывать эту переменную. У S7-1200 CPU это 8 кБ.

Выходные параметры: STATUS_x, RED_ERR_S7, RED_ERR_DEV и TOT_COM_ERR

CPU отображает сообщения об ошибках на статусных выходах инструкции MB_RED_CLIENT :

Примечание

*Коды ошибок могут отображаться в редакторе текстов программ в виде целочисленных значений или шестнадцатеричных значений.

1. Открыть требуемый блок в редакторе текстов программ.
2. Для включения состояние программирования нажать кнопку "Вкл/выкл мониторинг". (Если онлайн-соединение отсутствует, откроется диалоговое окно "Установить онлайн-соединение". В этом диалоговом окне можно установить онлайн-соединение.)
3. Выбрать переменную (тег) для мониторинга, и выбрать в контекстном меню в пункте "Формат отображения" требуемый формат индикации.

- Параметр STATUS_x (общая информация о состоянии):

STATUS (W#16#)	Описание
0000	Инструкция была выполнена без ошибок.
0001	Соединение установлено.
0003	Соединение разорвано.
0A90	Инструкция MB_RED_CLIENT не лицензирована Более подробная информация содержится в разделе "Лицензирование" выше.
0AFF	Соединение не сконфигурировано и не используется. Соединение "0A" должно быть сконфигурировано.
7000	Нет активной операции и нет установленного соединения (REQ = 0, DISCONNECT = 1).
7001	Запушено установления соединения.
7002	Промежуточный вызов. Соединение устанавливается.
7003	Соединение завершается.
7004	Соединение установлено и контролируется. Нет активных обрабатываемых заданий.
7005	Идет отправка данных.
7006	Выполняется прием данных.

- Параметр STATUS_x (ошибка протокола)

STATUS (W#16#)	Описание
80C8	Отсутствие ответа от сервера за заданный промежуток времени. Проверить соединение с сервером Modbus. Об этой ошибке сообщается только по завершении сконфигурированных повторных попыток. Если инструкция MB_RED_CLIENT за установленный интервал времени не получит ответ с переданным ранее ID транзакции (см. статическую переменную (тег) MB_TRANSACTION_ID), то возвращается этот код ошибки.
8380	Принятая телеграмма Modbus имеет неправильный формат, или было получено слишком мало байтов.
8382	• Длина телеграммы Modbus в заголовке телеграммы не соответствует числу полученных байтов. • Число байтов не соответствует фактическому числу переданных байтов (только функции 1-4). • Начальный адрес в полученной телеграмме не соответствует сохраненному начальному адресу (функции 5, 6, 15 и 16). • Число слов не соответствует фактическому числу переданных слов (функции 15 и 16).
8383	Ошибка при чтении или записи данных или доступ вне диапазона адресов MB_DATA_PTR. Более подробная информация содержится в разделе "MB_DATA_PTR" выше.
8384	• Принят недействительный код исключительного условия. • Было принятое отличное от первоначально переданного клиентом значения данных (функции 5, 6 und 8) • Получено недействительное значение состояния (функция 11)

STATUS (W#16#)	Описание
8385	• Диагностический код не поддерживается. • Был принят отличный от первоначально переданного клиентом код подфункции (функция 8).
8386	Полученный код функции не соответствует изначально отправленному коду.
8387	Идентификатор протокола телеграммы Modbus TCP, полученной сервером, не равен "0".
8388	Длина данных, отправленных сервером Modbus, отличается от обработанной. Эта ошибка возникает только при использовании функций Modbus 5, 6, 15 или 16.

- Параметр STATUS_x (ошибка параметра)

STATUS (W#16#)	Описание
80B6	Недействительный тип соединения; поддерживаются только TCP-соединения.
80BB	Неверное значение параметра ActiveEstablished. Для клиента разрешена только активная установка соединения (ActiveEstablished = TRUE).
8188	Неверное значение параметра MB_MODE.
8189	Неправильная адресация данных в параметре MB_DATA_ADDR
818A	Недействительная длина данных в параметре MB_DATA_LEN.
818B	Неверный указатель параметра MB_DATA_PTR. Необходимо проверить и значения параметров MB_DATA_ADDR и MB_DATA_LEN. (Дополнительная информация о "MB_DATA_ADDR" см. в разделе "MB_DATA_ADDR" выше.)
818C	Превышение времени в параметре BLOCKED_PROC_TIMEOUT или RCV_TIMEOUT (см. статические переменные (теги) инструкции). BLOCKED_PROC_TIMEOUT иRCV_TIMEOUT должны лежать в диапазоне между 0,5 с и 55,0 с.
8200	• В настоящее время CPU обрабатывает на порте другой запрос Modbus. • Другой экземпляр MB_RED_CLIENT с такими же параметрами соединения обрабатывается активный запрос Modbus.

Примечание

Коды ошибок используемых системой коммуникационных инструкций

С инструкцией MB_RED_CLIENT в дополнение к ошибкам, перечисленным в таблицах, могут возникать ошибки, вызванные коммуникационными инструкциями (TCON, TDISCON, TSEND, TRCV, T_DIAG и TRESET), используемыми инструкцией.

CPU назначает коды ошибок через блок данных экземпляра инструкции MB_RED_CLIENT. CPU отображает коды ошибок для соответствующей инструкции в разделе "Static" в STATUS.

Значение кодов ошибок доступно в документации для соответствующей коммуникационной инструкции.

Примечание

Ошибки коммуникации при передаче или приеме данных

Если при передаче или приеме данных возникает ошибка коммуникации (80C4 (Временная ошибка коммуникации. Указанное соединение временно прервано.), 80C5 (Удаленный партнер активно разорвал соединение.) или 80A1 (Указанное соединение было разорвано или еще не установлено.)), то CPU разрывает существующее соединение.

Это означает, что можно видеть все возвращенные значения STATUS, когда соединение разрывается, и что код STATUS, который вызвал отмену соединение, выводится только тогда, когда соединение разрывается.

Пример: Если при получении данных возникает временная ошибка коммуникации, то сначала выводится STATUS 7003 (ERROR = false), а затем 80C4 (ERROR = true).

MB_RED_SERVER (коммуникация через PROFINET в качестве сервера Modbus TCP)

Можно использовать эту инструкцию для создания соединения между S7-1200 CPU и устройством с поддержкой протокола MODBUS TCP.

Таблица 13-73 Инструкция MB_RED_SERVER

LAD/FBD	SCL	Описание

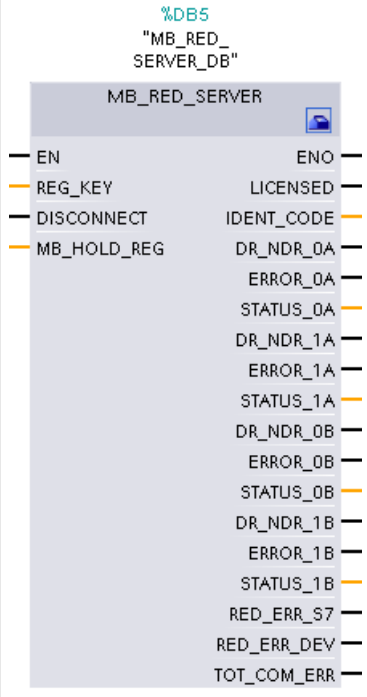
	`"MB_RED_SERVER_DB"(`
	`  DISCONNECT:=_bool_in_,`
	`  LICENSED=>_bool_out_`
	`  IDENT_CODE=>_string_out_`
	`  DR_NDR_0A=>_bool_out,`
	`  ERROR_0A=>_bool_out,`
	`  STATUS_0A=>_word_out_,`
	`  DR_NDR_1A=>_bool_out,`
	`  ERROR_1A=>_bool_out,`
	`  STATUS_1A=>_word_out_,`
	`  DR_NDR_0B=>_bool_out,`
	`  ERROR_0B=>_bool_out,`
	`  STATUS_0B=>_word_out_,`
	`  DR_NDR_1B=>_bool_out,`
	`  ERROR_1B=>_bool_out,`
	`  STATUS_1B=>_word_out_,`
	`  RED_ERR_S7=>_bool_out_,`
	`  RED_ERR_DEV=>_bool_out_,`
	`  TOT_COM_ERR=>_bool_out_,`
	`  MB_HOLD_REG:=_variant_inout_);`
		Инструкция MB_RED_SERVER выполняет обмен данными в качестве сервера Modbus TCP через соединение PROFINET.
		Инструкция MB_RED_SERVER обрабатывает запросы на соединение от клиентов Modbus TCP, принимает и обрабатывает запросы Modbus и передает ответы.

Таблица 13-74 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REG_KEY [1]	IN	STRING[17]	Регистрационный код для лицензирования Инструкция MB_RED_SERVER должна лицензироваться индивидуально на каждом CPU.
DISCONNECT	IN	Bool	С помощью инструкции MB_RED_SERVER выполняется вход в пассивное соединение с партнерским модулем. Сервер реагирует на запрос соединения с IP-адресов, указанных в описаниях соединений как специфицированные или не специфицированные. С помощью этого параметра выбирается, когда будет принят запрос на соединение: • 0: CPU устанавливает пассивное соединение, если коммуникационное соединение отсутствует. • 1: Инициализация разрыва соединения. Если вход установлен, то CPU не обрабатывает дополнительные клиентские запросы и запускается разрыв соединения. После успешного разрыва соединения в параметре STATUS_x возвращается значение "0003".
MB_HOLD_REG [2]	IN_OUT	Variant	Указатель на регистр хранения Modbus инструкции MB_RED_SERVER MB_HOLD_REG всегда должен указывать на область памяти с размером больше двух байт. Регистр хранения содержит значения, к которым клиент Modbus может обращаться с помощью функций Modbus 3 (чтение), 6 (запись), 16 (многократная запись) и 23 (чтение и запись в одном задании).
LICENSED [1]	OUT	Bool	• 0: Инструкция не лицензирована • 1: Инструкция лицензирована
IDENT_CODE [1]	OUT	STRING[18]	Идентификация для лицензирования. Использовать следующую строку для запроса регистрационного кода REG_KEY.
DR_NDR_0A	OUT	Bool	"Data Read" или "New Data Ready" для соединения 0A: • 0: Новых данных нет • 1: Новые данные, прочитанные или записанные клиентом Modbus
ERROR_0A	OUT	Bool	Если при вызове инструкции MB_RED_SERVER для соединения 0A возникает ошибка, то выход параметра ERROR_0A устанавливается на "1". Подробная информация о причине ошибки содержится в параметре STATUS_0A.
STATUS_0A [3]	OUT	Word	Подробная информация о состоянии инструкции по соединению 0A.
DR_NDR_1A	OUT	Bool	"Data Read" или "New Data Ready" для соединения 1A: • 0: Новых данных нет • 1: Новые данные, прочитанные или записанные клиентом Modbus
ERROR_1A	OUT	Bool	Если при вызове инструкции MB_RED_SERVER для соединения 1A возникает ошибка, то выход параметра ERROR_1A устанавливается на "1". Подробная информация о причине ошибки содержится в параметре STATUS_1A.
STATUS_1A [3]	OUT	Word	Подробная информация о состоянии инструкции по соединению 1A.
DR_NDR_0B	OUT	Bool	"Data Read" или "New Data Ready" для соединения 0B: • 0: Новых данных нет • 1: Новые данные, прочитанные или записанные клиентом Modbus

Параметр и тип данных		Тип данных	Описание
ERROR_0B	OUT	Bool	Если при вызове инструкции MB_RED_SERVER для соединения 0B возникает ошибка, то выход параметра ERROR_0B устанавливается на "1". Подробная информация о причине ошибки содержится в параметре STATUS_0B.
STATUS_0B [3]	OUT	Word	Подробная информация о состоянии инструкции по соединению 0B.
DR_NDR_1B	OUT	Bool	"Data Read" или "New Data Ready" для соединения 1B: • 0: Новых данных нет • 1: Новые данные, прочитанные или записанные клиентом Modbus
ERROR_1B	OUT	Bool	Если при вызове инструкции MB_RED_SERVER для соединения 1B возникает ошибка, то выход параметра ERROR_1B устанавливается на "1". Подробная информация о причине ошибки содержится в параметре STATUS_1B.
STATUS_1B [3]	OUT	Word	Подробная информация о состоянии инструкции по соединению 1B.
RED_ERR_S7 [3]	OUT	Bool	• 0: Ошибки резервирования в SIMATIC отсутствуют • 1: Ошибка резервирования в SIMATIC
RED_ERR_S7 [3]	OUT	Bool	• 0: Ошибки резервирования на стороне партнёра по соединению отсутствуют • 1: Ошибка резервирования на стороне партнёра по соединению
RED_ERR_S7 [3]	OUT	Bool	• 0: Как минимум 1 сконфигурированное соединение установлено • 1: Полная потеря связи, все сконфигурированные соединения разорваны

[1] Более подробная информация содержится в следующем разделе "Лицензирование".

[2] Более подробная информация содержится в разделе "Входной параметр MB_HOLD_REG" ниже.

[3] Более подробная информация содержится в следующем разделе "Выходные параметры:ERROR_x, RED_ERR_S7, RED_ERR_DEV и TOT_COM_ERR".

Примечание

Указания по безопасности

Каждый клиент в сети имеет доступ на чтение и запись к входам и выходам образа процесса, а также к области блоков данных и битовой памяти, которые определены в регистре хранения Modbus. Эта опция должна ограничить доступ к IP-адресу для предотвращения несанкционированных операций чтения и записи. Однако следует обратить внимание на то, что общий адрес может быть использован и для несанкционированного доступа.

Примечание

Требования к версии прошивки CPU

Инструкции Modbus TCP, описанные в этом разделе руководства, требуют прошивки версии V4.2 или выше.

Для использования инструкции дополнительный аппаратный модуль не требуется.

Подключения нескольких серверов

CPU могут:

- Обрабатывать подключения нескольких серверов
- Принимать несколько подключений от разных клиентов одновременно на одном порту сервера

Макс. число соединений зависит от используемого CPU и указано в его технических параметрах. Общее количество соединений CPU, включая клиентов и серверы Modbus TCP, не должно превышать максимальное количество поддерживаемых соединений.

Для серверных соединений необходимо учитывать следующие правила:

- Каждое соединение MB_RED_SERVER должно использовать уникальный DB экземпляра.
- Для каждого клиента, который хочет подключиться к порту сервера, требуется уникальное соединение/идентификатор соединения.
- Идентификаторы соединений также должны быть однозначными в пределах ЦПУ.

Согласование адресов Modbus для образа процесса

Инструкция MB_RED_SERVER позволяет входящим функциям Modbus (1, 2, 4, 5 и 15) напрямую выполнять чтение и запись на входы и выходы образа процесса CPU (использование типов данных BOOL и WORD).

Для S7-1200 CPU адресное пространство для образа процесса входов и образа процесса выходов составляет 1 кБ.

В следующей таблице показано адресное пространство перечисленных выше функций Modbus:

Функция Modbus					
Код функции	**Функция**	**Область данных**	**Адресное пространство**		
01	Чтение: Биты	Output	0	до	65.535
02	Чтение: Биты	Input	0	до	65.535
04	Чтение: WORD	Input	0	до	65.535
05	Запись: Бит	Output	0	до	65.535
15	Запись: Биты	Output	0	до	65.535

Входящие запросы Modbus к кодами функций 3, 6, 16 и 23 записывают или считывают регистр хранения Modbus (регистр хранения определяется с помощью параметра MB_HOLD_REG или через Data_Area_Array).

Функции Modbus

В таблице ниже перечислены все функции Modbus, поддерживаемые инструкцией MB_RED_SERVER:

Код функции	Описание
01	Считывание выходных битов
02	Считывание входных битов
03	Считывание регистра хранения
04	Считывание входных слов
05	Запись выходного бита
06	Запись регистра хранения
08	Функции диагностики: • эхо-тест (подфункция 0x0000): Инструкция MB_RED_SERVER принимает слово данных и возвращает его в неизменном виде клиенту Modbus. • сброс счетчика событий (подфункция 0x000A): Инструкция MB_RED_SERVER сбрасывает следующие счетчики событий: "Success_Count", "Xmt_Rcv_Count", "Exception_Count", "Server_Message_Count" и "Request_Count".
11	Функции диагностики: Получение значений от счетчика событий коммуникации Инструкция MB_RED_SERVER использует внутренний счетчик событий коммуникации для регистрации количества выполненных запросов чтения и записи, которые отправляются на сервер Modbus. Значение счетчика не увеличивается при выполнении функций 8 и 11. Это же относится и к запросам, ставшим причиной ошибки коммуникации, напр., при возникновении ошибки протокола. Код функции в принятом запросе Modbus не поддерживается.
15	Считывание выходных битов
16	Запись регистра хранения
23	С запросом на запись регистра хранения и чтение регистра хранения

Эксплуатация и резервирование

Участники обмена данными могут быть выполнены как автономные или резервируемые узлы. Если один из партнеров является автономным, то речь идет об одностороннем резервировании. Если оба партнера являются резервируемыми, то речь идет о двустороннем резервировании:

- Одностороннее резервирование:
  - Для каждого соединения между партнерами должна быть сконфигурирована отдельная линия. Точки подсоединения **SIMATIC S7** обозначаются как **0** и **1**. Точки подсоединения **партнеров** обозначаются как **A** и **B**.
    R-CPU или H-CPU 1 относится к точке подсоединения 0, R-CPU или H-CPU 2 относится к точке подсоединения 1.
  - Конфигурация: В резервируемой конфигурации S7 создается соединение S7 точки подсоединения 0 с узловой точкой A партнёра по соединению (подключение S7 точки подсоединения **0** к партнеру/узлу **A** => соединение **0A**), и создается соединение S7 точки подсоединения 1 с узловой точкой A партнёра по соединению (подключение S7 точки подсоединения **1** к партнеру/узлу **A** => соединение **1A**). На рисунке показаны обозначения соединения.

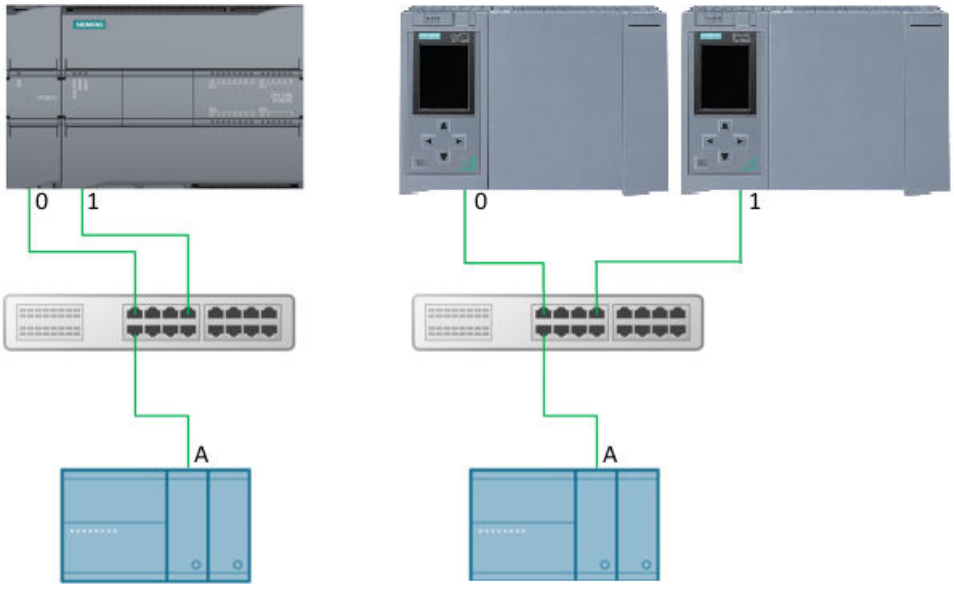


Рисунок 13-12 Одностороннее резервирование S7

– Если S7 является автономным, а партнёр по соединению резервируемым, то создается соединение S7 точки подсоединения 0 с узловой точкой A партнёра по соединению (подключение S7 точки подсоединения **0** к партнеру/узлу **A** => соединение **0A**), и создается соединение S7 точки подсоединения 0 с узловой точкой B партнёра по соединению (подключение S7 точки подсоединения **0** к партнеру/узлу **B** => соединение **0B**). На рисунке показаны обозначения соединения.

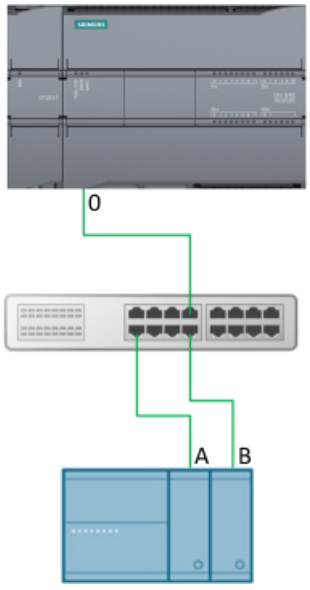


Рисунок 13-13 Одностороннее резервирование, партнер

- Двухстороннее резервирование:
  – Описание: Для каждого соединения между партнерами должна быть сконфигурирована отдельная линия. Точки подсоединения **SIMATIC S7** обозначаются как 0 и **1**. Точки подсоединения **партнеров** обозначаются как **A** и **B**. R-CPU или H-CPU 1 относится к точке подсоединения 0, R-CPU или H-CPU 2 относится к точке подсоединения 1.

– Конфигурация: При двухстороннем резервировании создается два подключения с началом в точке подсоединения 0 (подключение S7 точки подсоединения **0** к парнеру/узлу **A** => соединение **0A** и подключение S7 точки подсоединения**0** к парнеру/узлу **B** => соединение **0B**), а также создается два подключения с началом в S7 точке подсоединения 1 к узловым точкам A и B партнёра по соединению (подключение S7 точки подсоединения **1** к парнеру/узлу **A** => соединение **1A** и подключение S7 точки подсоединения **1** к парнеру/узлу **B** => соединение **1B**). На рисунке показаны обозначения соединения.

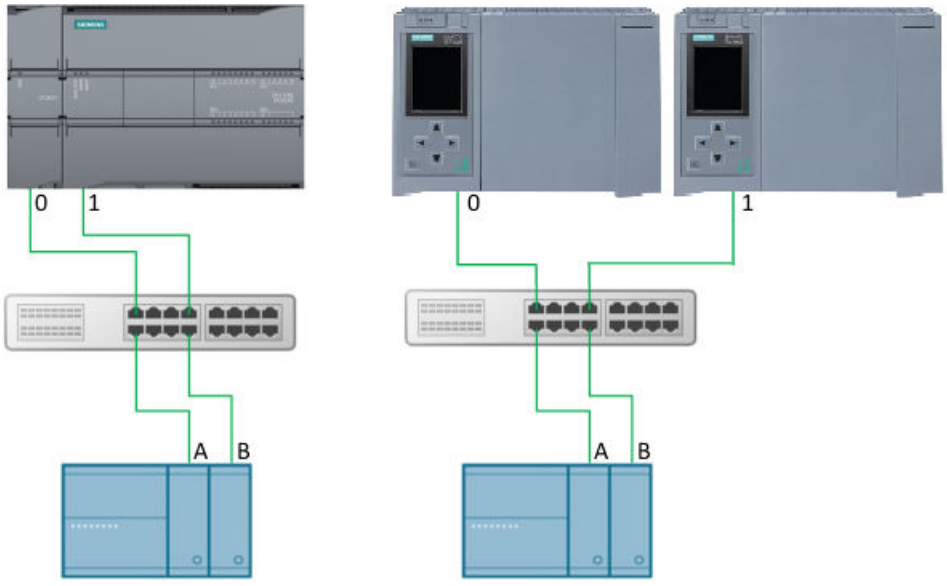


Рисунок 13-14 Двухстороннее резервирование:

- Обработка телеграмм: Телеграммы могут приниматься по всем сконфигурированным соединениям. Клиент может отправлять телеграммы по одному или по всем соединениям. После поступления телеграммы на соединение, CPU показывает состояние на соответствующем выходе DR_NDR_x или ERROR_x. Каждое соединение является автономным и не влияет на отображение других соединений.
- Выходы резервирования RED_ERR_S7, RED_ERR_DEV и TOT_COM_ERR:
  - Биты резервирования RED_ERR_S7, RED_ERR_DEV и TOT_COM_ERR устанавливаются в зависимости от статуса выходов состояния:

Number of faulty connections	STATUS_0A	STATUS_0B	STATUS_1A	STATUS_1B	RED_ERR_S7	RED_ERR_DEV	TOT_COM_ERR
0	okay	okay	okay	okay	FALSE	FALSE	FALSE
1	okay	okay	okay	Error	FALSE	FALSE	FALSE
	okay	okay	Error	okay	FALSE	FALSE	FALSE
	okay	Error	okay	okay	FALSE	FALSE	FALSE
	Error	okay	okay	okay	FALSE	FALSE	FALSE
2	okay	okay	Error	Error	TRUE	FALSE	FALSE
	okay	Error	okay	Error	FALSE	TRUE	FALSE
	Error	okay	okay	Error	FALSE	FALSE	FALSE
	okay	Error	Error	okay	FALSE	FALSE	FALSE
	Error	okay	Error	okay	FALSE	TRUE	FALSE
	Error	Error	okay	okay	TRUE	FALSE	FALSE
3	Error	Error	Error	okay	TRUE	TRUE	FALSE
	Error	Error	okay	Error	TRUE	TRUE	FALSE
	Error	okay	Error	Error	TRUE	TRUE	FALSE
	okay	Error	Error	Error	TRUE	TRUE	FALSE
4	Error	Error	Error	Error	TRUE	TRUE	TRUE

Рисунок 13-15 Отображение битов прерывания для установки резервирования с обеих сторон

Number of faulty connections	STATUS_0A	STATUS_0B	STATUS_1A	STATUS_1B	RED_ERR_S7	RED_ERR_DEV	TOT_COM_ERR
0	okay	0AFF	okay	0AFF	FALSE	FALSE	FALSE
1	okay	0AFF	Error	0AFF	TRUE	TRUE	FALSE
	Error	0AFF	okay	0AFF	TRUE	TRUE	FALSE
2	Error	0AFF	Error	0AFF	TRUE	TRUE	TRUE

Рисунок 13-16 Отображение битов прерывания для установки резервирования с одной сторон

Примечание

Номера портов для клиента и сервера

Modbus клиент используется номер порта, начинающий на 2000. Для обращения к Modbus серверу обычно используется порт с номером 502. В зависимости от CPU, можно сконфигурировать порт 502 для нескольких соединений (мультипорт). Если локальный порт 502 был сконфигурирован для двух или более соединений, обращающиеся клиенты в случайном порядке распределяются по существующим серверным соединениям, если таковые не определены. Первому клиенту, пожелавшему установить соединение с инструкцией "MB_RED_SERVER", соединение 0A не присваивается автоматически. После назначения клиентских запросов серверным соединениям, назначение сохраняется на время обмена телеграммами до завершения соединения.

Параметрирование

Для S7-1200 можно использовать инструкцию MB_RED_SERVER **V1.0** и **V1.1**. CPU реализует соединения через локальный интерфейс CPU или CM/CP. Через структуру TCON_IP_V4 соединения конфигурируются CPU и устанавливаются.

Конфигурация MB_RED_SERVER: Следующие установки выполняются в диалоговом окне конфигурирования для инструкции MB_RED_SERVER:

- Параметры для соединений 0A, 1A, 0B и 1B (дополнительную информацию о конфигурации резервирования можно найти в разделе "Эксплуатация и резервирование" выше)
- Внутренние параметры (необязательно)

Для открытия диалогового окна конфигурирования можно использовать инструкцию MB_RED_SERVER или технологические объекты:

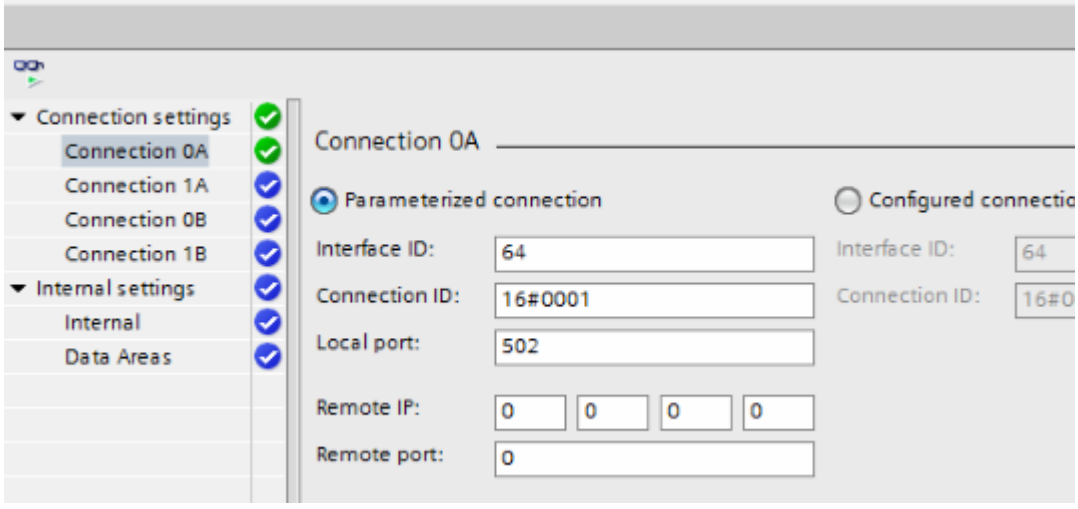


Рисунок 13-17 Настроенные серверные соединения

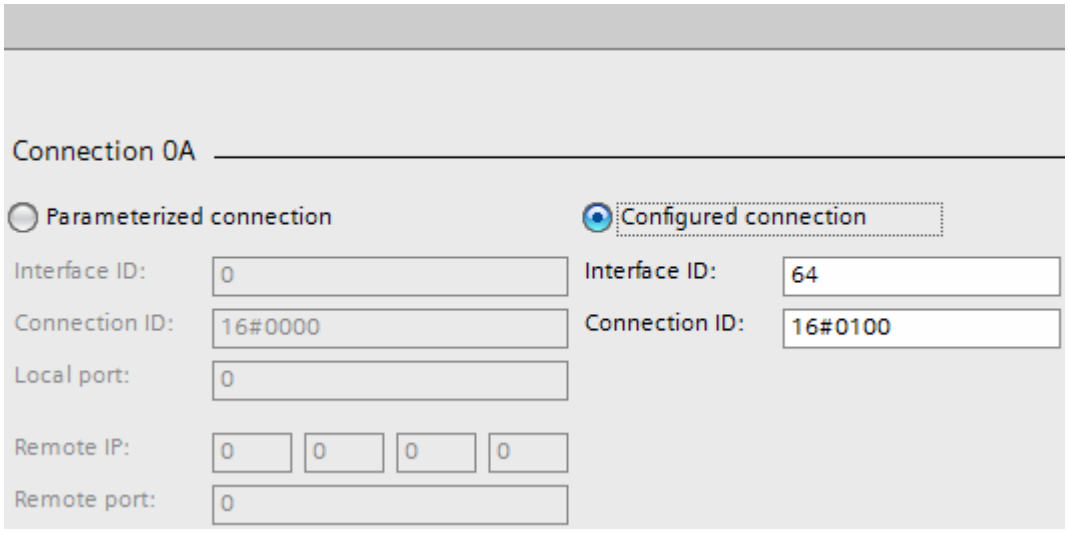


Рисунок 13-18 Сконфигурированное серверное соединение

Переменная (тег)	Стартовое значение	Описание
Сконфигурированные соединения		
Interface ID	64	Аппаратный идентификатор используемого PN-интерфейса
Connection ID	16#0000	ID соединений для используемых соединений. Эти идентификаторы соединений должны быть однозначными в пределах CPU.
Local port	502	Номер локального порта блока сервера. Порт по умолчанию для Modbus/TCP сервера 502.
Remote IP	0.0.0.0	Удаленный IP-адрес клиента. По умолчанию IP-адрес для клиента не введен.
Remote port	0	Удаленный номер порта клиента. По умолчанию номер порта для клиента не введен.
Сконфигурированные соединения		
Interface ID	64	Аппаратный идентификатор используемого PN-интерфейса
Connection ID	16#0000	ID соединений для используемых соединений. Эти соединения конфигурируются в обзоре сетей.

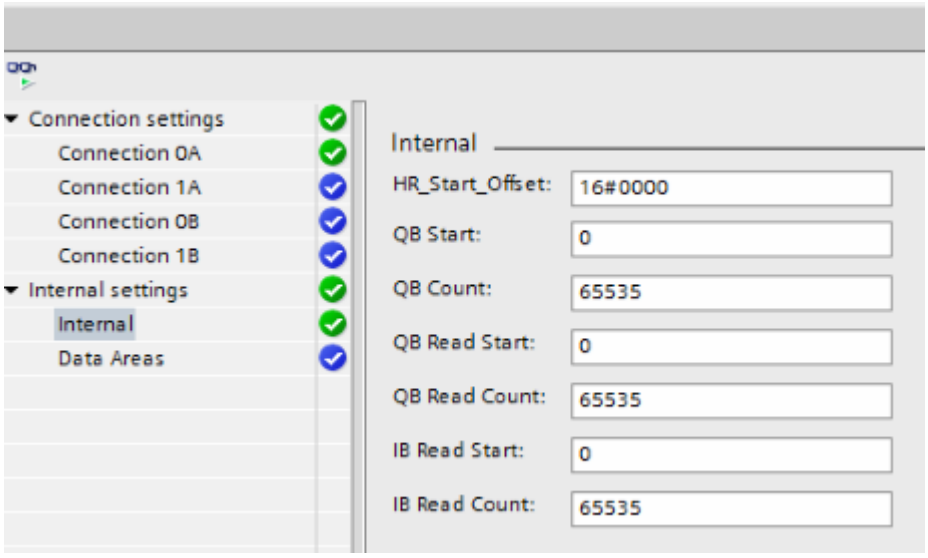


Рисунок 13-19 Внутренние параметры (необязательно)

Переменная (тег)	**Тип данных**	**Стартовое значение**	**Описание**
HR_Start_Offset	WORD	0	Присвоение начального адреса регистра хранения Modbus.
QB_Start	UINT	0	Начальный адрес разрешенного диапазона адресов выходов, в которые может записывать ведущий Modbus (байты от 0 до 65535)
QB_Count	UINT	0	Число выходных байтов, в которые может записывать ведущий Modbus *Пример:* • QB_Start = 0 и QB_Count = 10: Ведущее устройство Modbus может выполнять запись в выходные байты от 0 до 9. • QB_Count = 0: Ведущее устройство Modbus не может выполнять запись в выходные байты.
QB_Read_Start	UINT	0	Начальный адрес разрешенного диапазона адресов для выходов, которые может читать ведущий Modbus (байты от 0 до 65535)
QB_Read_Count	UINT	0	Число выходных байтов, в которые может читать ведущий Modbus *Пример:* • QB_Read_Start = 0 и QB_Read_Count = 10: Ведущее устройство Modbus может выполнять чтение выходных байт от 0 до 9. • QB_Read_Count = 0: Ведущее устройство Modbus не может выполнять чтение выходных байтов.
IB_Read_Start	UINT	0	Начальный адрес разрешенного диапазона адресов для входов, которые может читать ведущий Modbus (байты от 0 до 65535)

Переменная (тег)	Тип данных	Стартовое значение	Описание
IB_Read_Count	UINT	0	Число входных байтов, в которые может читать ведущий Modbus *Пример:* IB_Read_Start = 0 и IB_Read_Count = 10: Ведущее устройство Modbus может выполнять чтение входных байтов от 0 до 9. IB_Read_Count = 0: Ведущее устройство Modbus не может выполнять чтение входных байтов.
Data_Area_Array	ARRAY [1..8]		
data_type	UINT	0	Тип данных: от 0 до 4
db	UINT	0	Номер блока данных
start	UINT	0	Первый адрес Modbus в блоке данных
length	UINT	0	Количество значений Modbus в блоке данных

Адресация с помощью статической переменной HR_Start_Offset

Адреса регистра хранения Modbus начинаются с 0.

Пример: Регистр хранения начинается в MW100 и имеет длину в 100 слов.

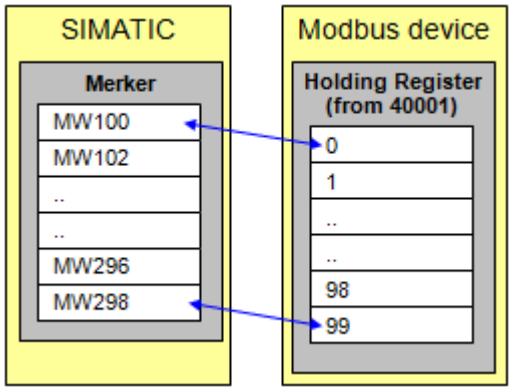


Рисунок 13-20 HR_Start_Offset_0

Переменную (тег) HR_Start_Offset можно задать и таким образом, чтобы регистр хранения Modbus получил начальный адрес, отличный от 0.

Пример: Значение 20 для смещения в параметре HR_Start_Offset означает, что начальный адрес регистра хранения смещается с 0 на 20. Это вызывает ошибку всякий раз при обращении регистру хранения ниже адреса 20 и выше адреса 119.

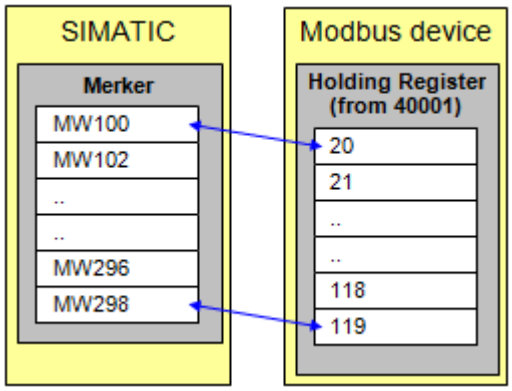


Рисунок 13-21 HR_Start_Offset_20

Data_Area_Array [1..8]

Для согласования адресов MODBUS в памяти SIMATIC S7 доступно восемь областей данных. Если область данных определяется с типом данных "Регистр хранения", параметр MB_HOLD_REG не обрабатывается. Вместо этого ведущее устройство Modbus записывает или считывает регистр Modbus и биты в блоках данных в зависимости от типа задания. CPU может дополнительно обрабатывать эти значения при последующем выполнении программы.

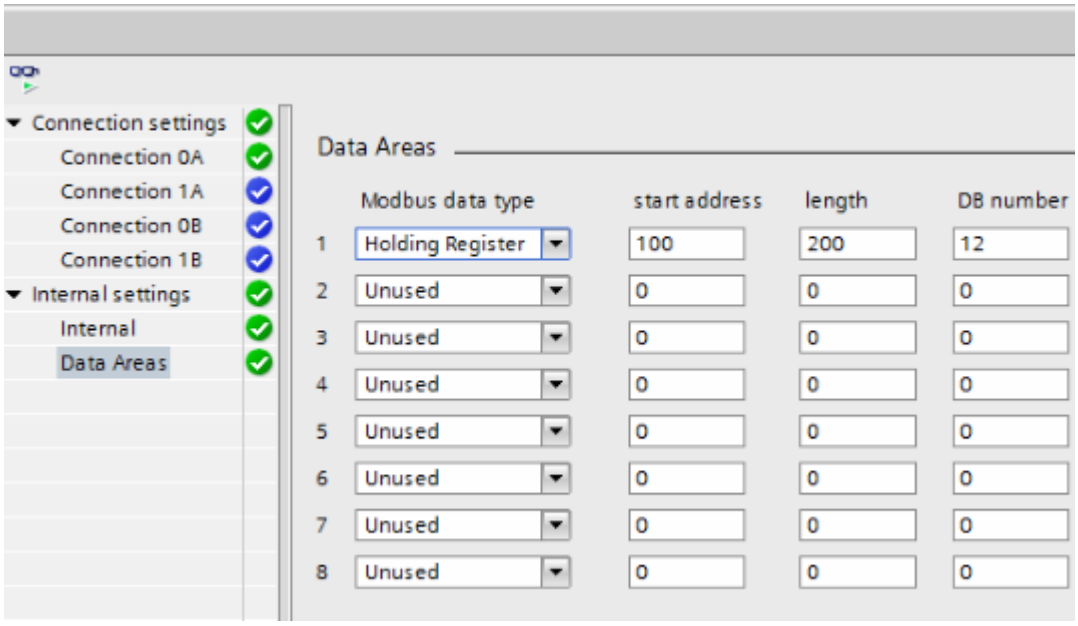


Рисунок 13-22 Области данных сервера

Одно задание позволяет читать только из одного DB или записывать в один DB. Доступ к регистрам или значениям битов, которые находятся в разных DB, даже если номера расположены последовательно и без пробелов, должен быть разделен на две операции. Это следует учитывать при конфигурировании. Можно согласовать больше областей Modbus (регистров или битовых значений) в одном блоке данных, чем ведущее устройство Modbus может обработать в одной телеграмме.

data_type

Параметр data_type определяет, какие типы данных MODBUS ведущее устройство Modbus согласует в этом блоке данных. Если в data_type вводится значение "0", то ведущее устройство Modbus не использует соответствующую область данных. Если ведущее устройство Modbus должен использовать несколько Data_Area, то они должны быть определены один за другим. Ведущее устройство Modbus не обрабатывает никаких записей последata_type = 0. .

Иден-тифи-катор	Тип данных	Описание
0	Диапазон не используется	
1	Выходные биты (катушки)	Бит
2	Входные биты (входы)	Бит
3	Регистр хранения	Word
4	Входные биты (входной регистр)	Word

db

Параметр db определяет блок данных, который согласует регистры MODBUS или битовые значения, определенные ниже. CPU не разрешает DB номер 0, потому что он зарезервирован для системы.

start, length

Параметр start определяет первый адрес Modbus, который ведущий Modbus согласует в слове данных 0 блока данных DB. Параметр length определяет длину, т.е. число адресов MODBUS, которые ведущее устройство Modbus согласует в блоке данных. Определенные области данных не должны перекрываться. Параметр length не должен быть 0.

Пример: Назначение адресов с Data_Area_Array

Область данных 1	data_type	3: Регистр хранения
	db	11
	start	0
	length	500
Область данных 2	data_type	3: Регистр хранения
	db	12
	start	720
	length	181
Область данных 3	data_type	4: Входные слова
	db	13
	start	720
	length	281
Область данных 4	data_type	1: Выходные биты
	db	14
	start	640
	length	611

Область данных 5	data_type	2: Входной бит
	db	15
	start	1700
	length	601
Область данных 6	data_type	1: Выходные биты
	db	16
	start	1700
	length	601
Область данных 7	data_type	Не используется
	db	0
	start	0
	length	0
Область данных 8	data_type	Не используется
	db	0
	start	0
	length	0

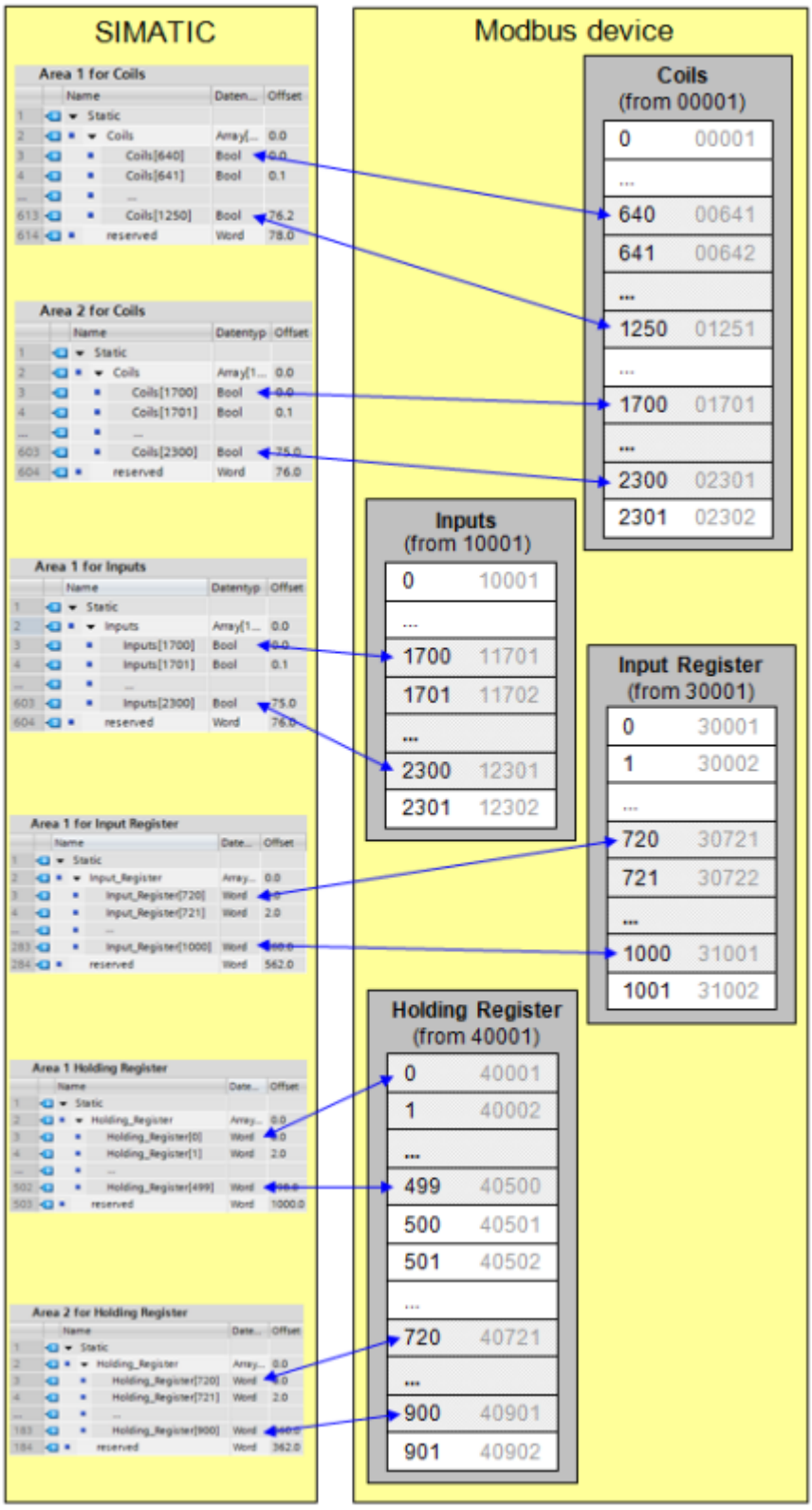


Рисунок 13-23 Структура адреса

Лицензирование

Инструкция MB_RED_SERVER является платной и подлежит лицензированию для каждого CPU по отдельности. Лицензирование выполняется за два этапа:

- Отображение IDENT_CODE лицензии
- Ввод регистрационного ключа REG_KEY: Необходимо ввести регистрационный ключ REG_KEY для каждой инструкции MB_RED_SERVER. Сохранить REG_KEY в глобальный блок данных, из которого все инструкции MB_RED_SERVER получают требуемый регистрационный ключ.

Порядок действий для отображения IDENT_CODE лицензии:

1. Задать параметры инструкции MB_RED_SERVER в соответствии с требованиями в циклическом OB. Загрузить программу в CPU и перевести CPU в RUN.
2. Открыть DB экземпляра Modbus инструкции и нажать кнопку "Показать все".
3. Экземплярный блок данных показывает на выходе IDENT_CODE 18-значную последовательность символов.

	Name	Data type	Start value	Monitor value
1	Input			
2	REG_KEY	String[17]	''	''
3	USE_ALL_CONN	Bool	false	FALSE
4	REQ	Bool	false	FALSE
5	DISCONNECT	Bool	false	FALSE
6	MB_MODE	USInt	0	0
7	MB_DATA_ADDR	UDInt	0	0
8	MB_DATA_LEN	UInt	0	0
9	Output			
10	LICENSED	Bool	false	FALSE
11	IDENT_CODE	String[18]	''	'RTPCFIGDCDIIHJHAH4'
12	DONE	Bool	false	FALSE
13	BUSY	Bool	false	FALSE
14	ERROR	Bool	false	FALSE

Рисунок 13-24 Лицензия

4. Скопировать эту последовательность символов с помощью функции Копировать/вставить из блока данных и вставить ее в форму (которая была получена после заказа продукта по электронной почте или находилась на CD).
5. Отправить форму через запрос на обслуживание на Служба поддержки клиентов (https://support.industry.siemens.com/my/ww/en/requests/#createRequest). В ответе придет регистрационный ключ для CPU.

Порядок действий для ввода регистрационного ключа REG_KEY:

1. Вставить через "Добавить новый блок..." новый глобальный блок данных с однозначным символьным именем, напр., "Лицензионный DB".
2. Создать в этом блоке параметр REG_KEY с типом данных STRING[17].

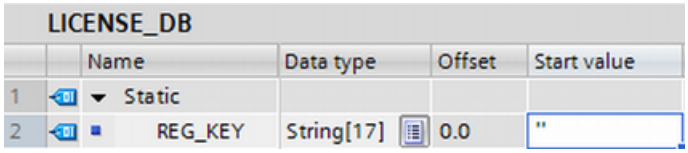


Рисунок 13-25 REG_KEY

3. Скопировать полученный 17-значный регистрационный ключ с помощью функции Копировать/вставить в столбец "Стартовое значение".
4. Ввести в циклическом OB в параметре REG_KEY инструкции MB_RED_SERVER имя лицензионного DB и имя последовательности символов (напр., License_DB.REG_KEY).
5. Загрузить измененные блоки в CPU. Регистрационный ключ может быть введен при работе. Перевод CPU из STOP в RUN не требуется.
6. Теперь Modbus/TCP коммуникация через инструкцию MB_RED_SERVER лицензирована для этого CPU, выходной бит LICENSED теперь TRUE.

Порядок действий для исправления отсутствующего или неправильного лицензирования:

- Если ввести неправильный регистрационный ключ или не вводить регистрационный ключ, то начинает мигать ERROR LED на CPU. Кроме этого, S7-1200 CPU выполняет циклическую запись в буфер диагностики с информацией об отсутствии лицензии.

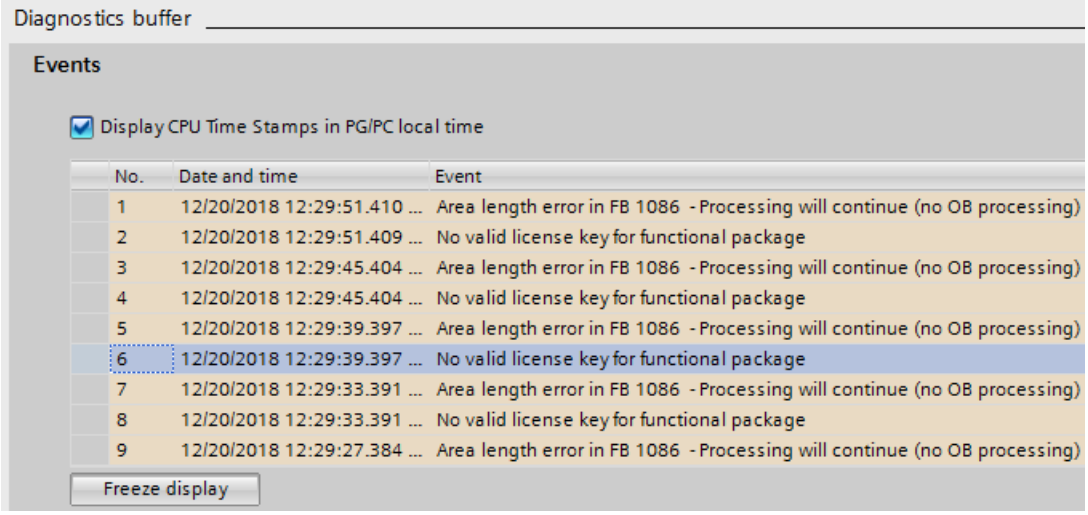


Рисунок 13-26 Буфер диагностики

- В случае отсутствующего или неправильного регистрационного ключа CPU продолжает обрабатывать коммуникацию Modbus TCP. Но на выходе STATUS_x непрерывно отображается "W#16#0A90" (нет действующего лицензионного ключа для пакета функций). Выходной бит LICENSED = FALSE.

Входной параметр MB_HOLD_REG

Параметр MB_HOLD_REG - это указатель на буфер данных для хранения данных, к которым клиент Modbus имеет доступ для чтения или записи. В качестве области памяти можно использовать глобальный блок данных (D) или битовую память (M):

- Верхний предел количества адресов в блоке данных (D) определяется максимальным размером DB в CPU.
- Верхний предел количества битовых запоминающих элементов (M) определяется максимальной областью битовой памяти в CPU.

На следующих рисунках показаны некоторые примеры согласования адресов Modbus для регистра хранения для функций Modbus 3 (чтение нескольких WORD), 6 (запись одного WORD), 16 (запись нескольких WORD) и 23 (запись и чтение нескольких WORD).

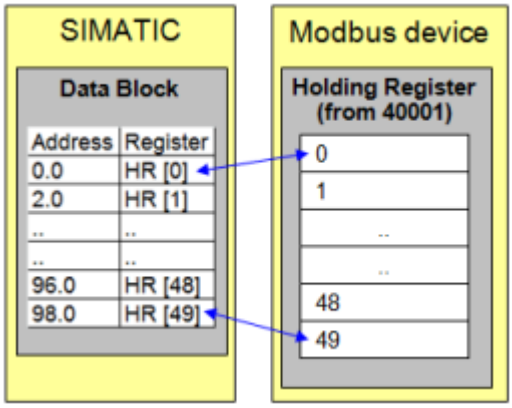


Рисунок 13-27 MB_HOLD_REG: Блок данных со смещением 0

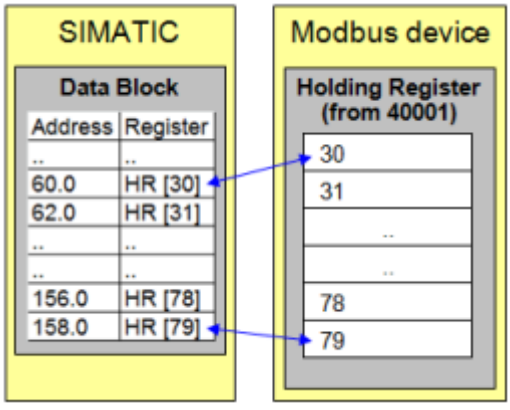


Рисунок 13-28 MB_HOLD_REG: Блок данных со смещением 60

Data_Area_Array [1..8]: Информацию об использовании опциональных параметров Data_Area_Array [1..8], можно найти в разделе "Параметрирование" выше.

Выходные параметры: ERROR_x, STATUS_x, RED_ERR_S7, RED_ERR_DEV и TOT_COM_ERR

CPU отображает сообщения об ошибках на статусных выходах инструкции MB_RED_SERVER:

Примечание

*Коды ошибок могут отображаться в редакторе текстов программ в виде целочисленных значений или шестнадцатеричных значений.

1. Открыть требуемый блок в редакторе текстов программ.
2. Для включения состояние программирования нажать кнопку "Вкл/выкл мониторинг". (Если онлайн-соединение отсутствует, откроется диалоговое окно "Установить онлайн-соединение". В этом диалоговом окне можно установить онлайн-соединение.)
3. Выбрать переменную (тег) для мониторинга, и выбрать в контекстном меню в пункте "Формат отображения" требуемый формат индикации.

Параметр STATUS_x (общая информация о состоянии):

STATUS (W#16#)	Описание
0000	Инструкция была выполнена без ошибок.
0001	Соединение установлено.
0003	Соединение разорвано.
0A90	Инструкция MB_RED_SERVER не лицензирована Более подробная информация содержится в разделе "Лицензирование" выше.
0AFF	Соединение не сконфигурировано и не используется. Соединение "0A" должно быть сконфигурировано.
7000	Нет активного вызова и нет установленного соединения (REQ = 0, DISCONNECT = 1).
7001	Первый вызов. Запушено установления соединения.
7002	Промежуточный вызов. Соединение устанавливается.
7003	Соединение завершается.
7005	Идет отправка данных.
7006	Выполняется прием данных.

Параметр STATUS_x (ошибка протокола)

STATUS (W#16#)	Код ошибки в сообщении об ошибке von MB_RED_SERVER(B#16#)	Описание
8380	-	Принятая телеграмма Modbus имеет неправильный формат, или было получено слишком мало байтов.
8381	01	Код функции не поддерживается.

STATUS (W#16#)	Код ошибки в сообщении об ошибке von MB_RED_SERVER(B#16#)	Описание
8382	03	Ошибка в длине данных: • Длина в принятой телеграмме Modbus указана неправильно • Длина телеграммы Modbus в заголовке телеграммы Modbus не соответствует числу фактически полученных байтов. • Число байтов в заголовке телеграммы Modbus не соответствует числу фактически полученных байтов (функции 15 и 16).
8383	02	Ошибка в адресе данных или доступ вне диапазона адресов регистра хранения (параметр MB_HOLD_REG). Более подробная информация содержится в разделе "MB_HOLD_REG" выше.
8384	03	Ошибка в значении данных (функция 05)
8385	03	Код диагностики не поддерживается (только для функции 08).

Параметр STATUS_x (ошибка параметра)

STATUS (W#16#)	Описание
80BB	Неверное значение параметра ActiveEstablished. Для сервера разрешено только пассивное установление соединения (active_established = FALSE).
8187	Неверный указатель параметра MB_HOLD_REG. Слишком маленькая область данных.
8389	Недействительное определение для области данных: • Недопустимое значение data_type • Номер DB недействителен или отсутствует: – Недопустимое значение db – Номер DB отсутствует – Номер DB уже использует в другой области данных – DB с оптимизированным доступом – DB находится не в рабочей памяти • Недопустимое значение length • Наложение диапазонов адресов MODBUS, относящихся к одному типу данных MODBUS

Примечание

Коды ошибок используемых системой коммуникационных инструкций

С инструкцией MB_RED_SERVER в дополнение к ошибкам, перечисленным в таблицах, могут возникать ошибки, вызванные коммуникационными инструкциями ("TCON", "TDISCON", "TSEND", "TRCV", "T_DIAG" и "T_RESET"), используемыми инструкцией.

Коды ошибок назначаются через блок данных экземпляра инструкции MB_RED_SERVER. Коды ошибок отображаются для соответствующей инструкции в разделе "Static" отдельных экземпляров в STATUS.

Значение кодов ошибок доступно в документации для соответствующей коммуникационной инструкции.

Примечание

Ошибки коммуникации при передаче или приеме данных

Если при передаче или приеме данных возникает ошибка коммуникации (80C4 (Временная ошибка коммуникации. Указанное соединение временно прервано.), 80C5 (Удаленный партнер активно разорвал соединение.), 80A1 (Указанное соединение было разорвано или еще не установлено.)), то CPU разрывает существующее соединение.

Это означает также, что можно видеть все возвращенные значения STATUS, когда соединение разрывается, и что код STATUS, который вызвал отмену соединение, выводится только тогда, когда соединение разрывается.

Пример: Если при получении данных возникает временная ошибка коммуникации, то сначала выводится STATUS 7003 (ERROR = false), а затем 80C4 (ERROR = true).

13.5.2.4 Примеры для Modbus TCP

Пример: MB_SERVER для нескольких соединений TCP

Возможно несколько соединений с сервером Modbus TCP. Для этого MB_SERVER должен выполняться независимо для каждого соединения. Каждое соединение должно использовать независимые экземплярный DB, ID соединения и IP порт. S7-1200 позволяет только одно соединение на IP-порт.

Для лучшей производительности MB_SERVER должен выполняться в каждом программном цикле для каждого соединения.

Параметр CONNECT использует системный тип данных TCP_IP_v4. Для примера эти структуры данных находятся в DB с именем "Соединения Modbus". DB "Соединения Modbus" содержит две структуры TCP_IP_v4: "TCPpassive_1" (для соединения 1) и "TCP_passive_2" (для соединения 2). Свойства соединения ID и LocalPort, описанные в комментариях к сегменту, являются элементами данных, сохраненными в структуре данных CONNECT.

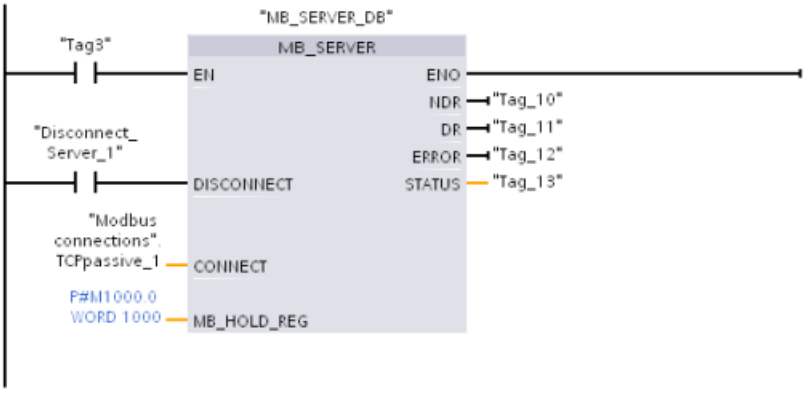
Данные TCP_IP_v4 CONNECT также содержат IP-адрес в массиве RemoteAddress ADDR . Назначения IP-адреса в TCPpassive_1 и TCP_passive_2 не влияют на установление соединений TCP сервера, но определяют, каким клиентам Modbus TCP разрешено связаться с каждым MB_SERVER. MB_SERVER пассивно контролирует Modbus на наличие

сообщений клиента и сравнивает IP-адрес входящего сообщения с IP-адресом, сохраненным в соответствующем массиве RemoteAddress ADDR.

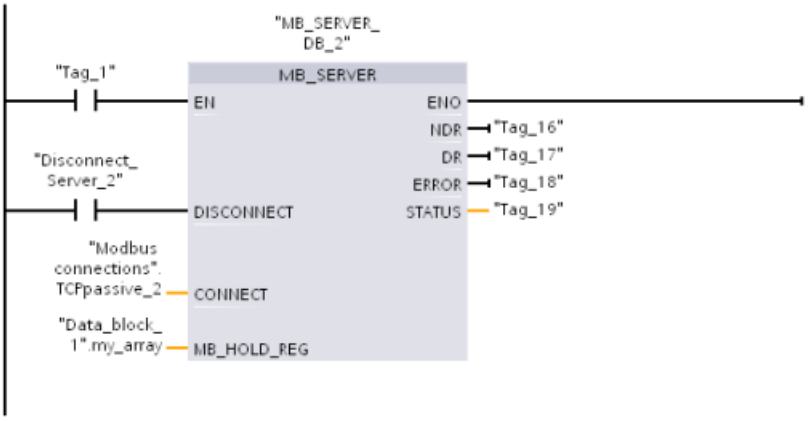
Три варианта IP-адреса возможны для двух инструкций MB_SERVER:

- **IP-адрес = 0.0.0.0**
  Каждая инструкция MB_SERVER ответит всем клиентам Modbus TCP, использующим любой IP-адрес.
- **IP-адрес = Один и тот же IP-адрес в TCPpassiv_1 и TCPpassiv_2**
  Оба соединения MB_SERVER отвечают только клиентам Modbus, обращающимся из этого IP-адреса.
- **IP-адрес = Разные IP-адреса в TCPpassiv_1 и TCPpassiv_2**
  Каждая инструкция MB_SERVER отвечает только клиентам Modbus, которые обращаются из IP-адреса, сохраненного в их данных TCP_IP_v4.

Сегмент 1: Соединение 1, DB экземпляра = "MB_SERVER_DB", в "Modbus connections.TCPpassiv_1" (ID = 1 и LocalPort = 502)



Сегмент 2: Соединение 2, DB экземпляра = "MB_SERVER_DB_1", в "Modbus connections.TCPpassiv_2" (ID = 2 и LocalPort = 503)



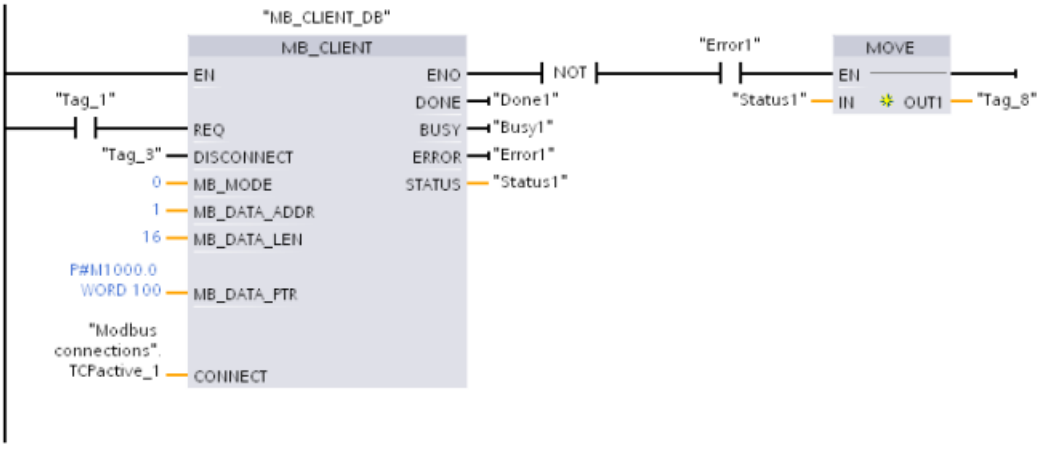
Пример: MB_CLIENT 1: Несколько запросов с общим TCP соединением

Несколько клиентских запросов Modbus могут быть отправлены по одному соединению. Для этого следует использовать тот же экземплярный DB , ID соединения и номер порта.

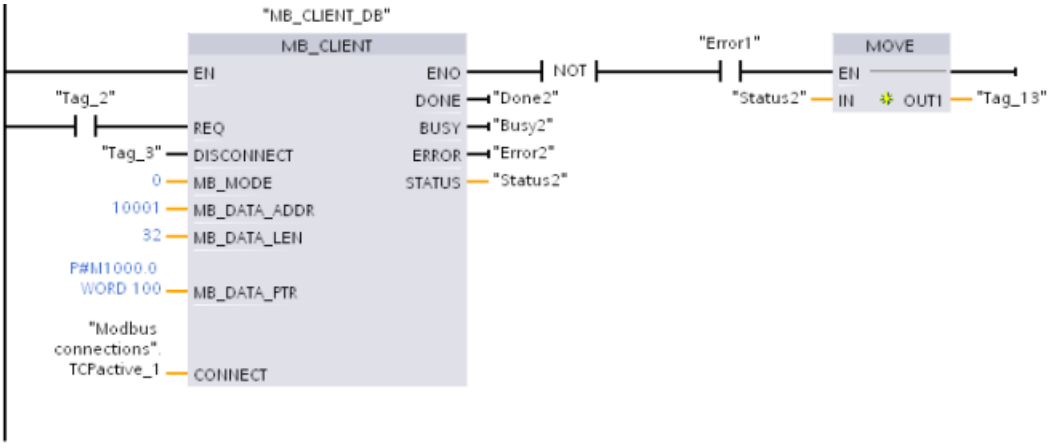
Поскольку оба блока MB_CLIENT используют ту же структуру данных TCON_IP_v4 в параметре CONNECT ("Modbus_connections".TCPactive_1), ID соединения, номер порта и IP-адрес совпадают. Данные IP-адреса CONNECT назначают IP-адрес целевого сервера Modbus TCP.

Только одна инструкция MB_CLIENT может быть активной в любой момент времени. Как только клиент завершает выполнение, следующий клиент может начать выполнение. Программная логика отвечает за последовательность выполнения. Пример демонстрирует обоих клиентов, считывающих удаленные данные из единственного клиента Modbus и передающих данные CPU клиента Modbus (M память, начиная с M1000.0). Возвращенная ошибка регистрируется, что является необязательным

Сегмент 1: Modbus функция 1 - считать 16 выходных битов из сервера Modbus TCP с IP-адресом, назначенным в "Modbus connections".TCPactive_1.



Сегмент 2: Modbus функция 2 - считать 32 входных бита из сервера Modbus TCP с IP-адресом, назначенным в "Modbus connections".TCPactive_1.



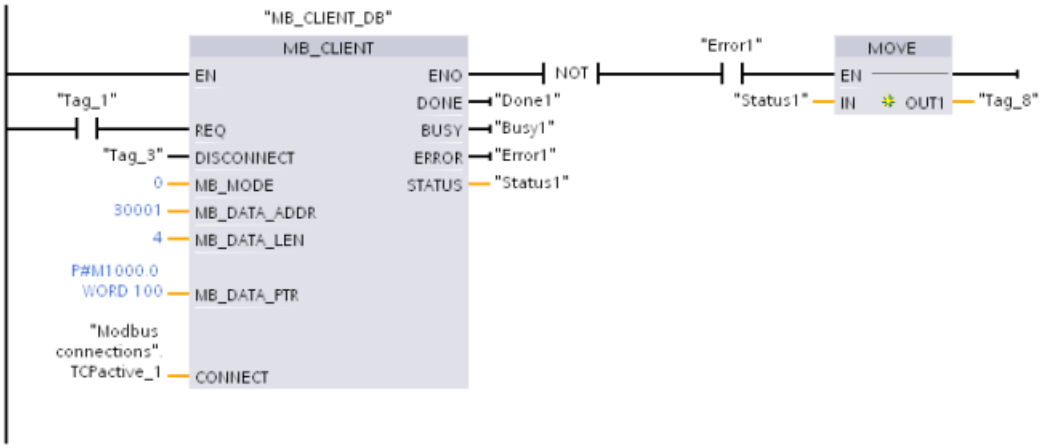
Пример: MB_CLIENT 2: Несколько запросов с различными TCP соединениями

Несколько клиентских запросов Modbus TCP могут быть отправлены по различным соединениям. Для этого должны использоваться различные экземплярные DB и ID соединения.

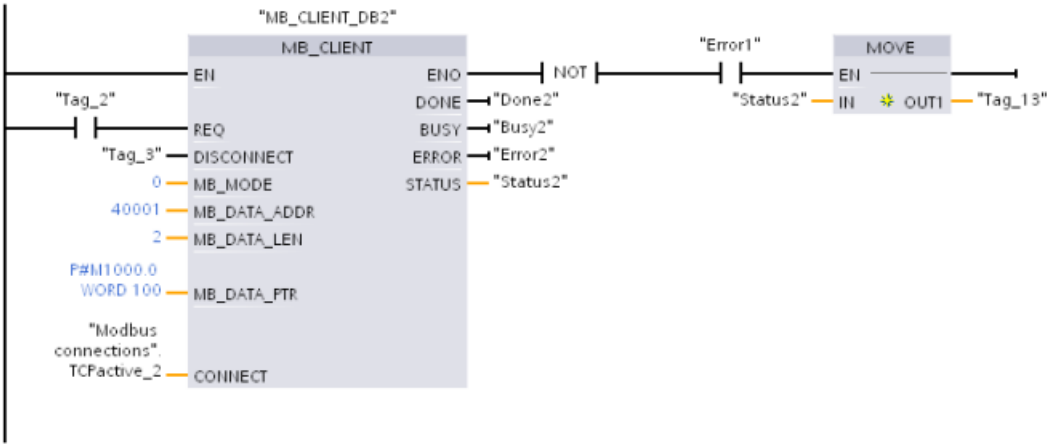
Номер RemotePort (IP-порт) должен отличаться, если соединения устанавливаются с тем же сервером Modbus. Если соединения находятся на различных серверах, нет никаких ограничений на номер IP-порта.

Пример демонстрирует двух клиентов Modbus TCP, передающих удаленные данные от двух различных серверов Modbus TCP в одну локальную область памяти CPU, начиная с адреса M1000.0. Кроме того, возвращенная ошибка регистрируется, что является необязательным.

Сегмент 1: Modbus функция 4 - Считать слова из образа процесса входов из сервера Modbus-TCP
Параметр CONNECT = "Modbus connections".TCPaktiv_1: **ID** соединения = 1, **RemoteAddress** = 192.168.2.241, **RemotePort** = 502



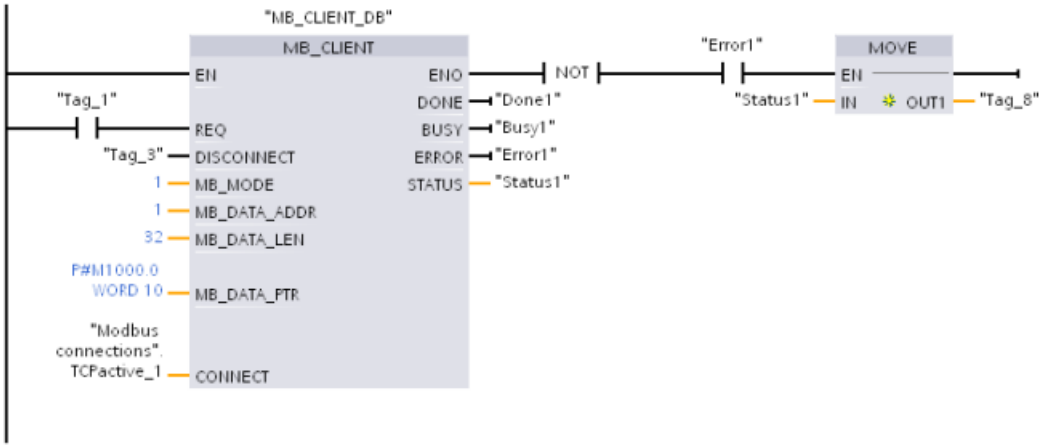
Сегмент 2: Modbus функция 3 - Считать слова из регистра хранения сервера Modbus-TCP
Параметр CONNECT = "Modbus connections".TCPaktiv_2: **ID** соединения = 2, **RemoteAddress** = 192.168.2.242, **RemotePort** = 502



Пример: MB_CLIENT 3: Запрос записи для образа процесса выходов

Этот пример показывает запрос клиента Modbus, который передает битовые данные из локальной памяти CPU (начиная с M1000.0) на удаленный сервер Modbus TCP.

Сегмент 1: Modbus функция 15 - Записать выходные биты на сервер Modbus

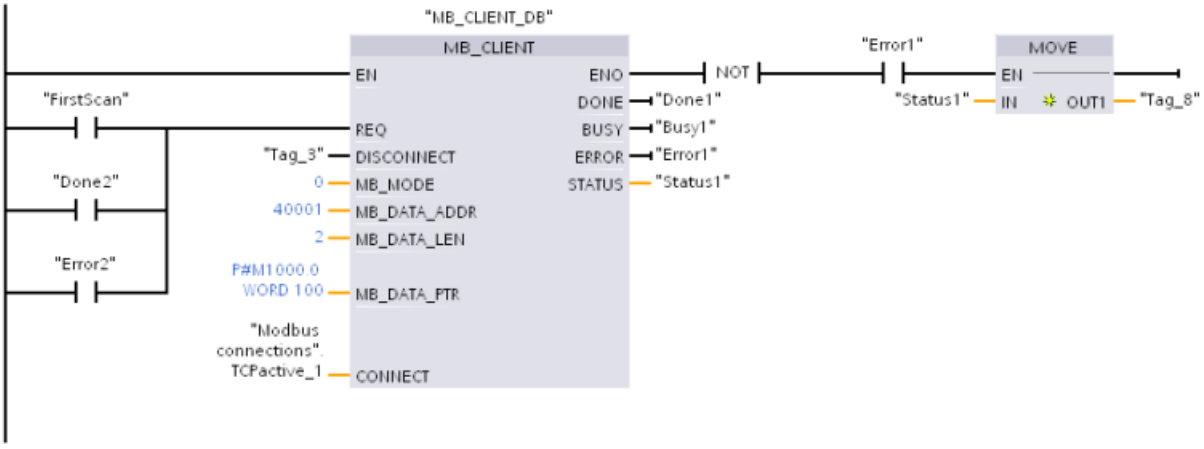


Пример: MB_CLIENT 4: Координация нескольких запросов

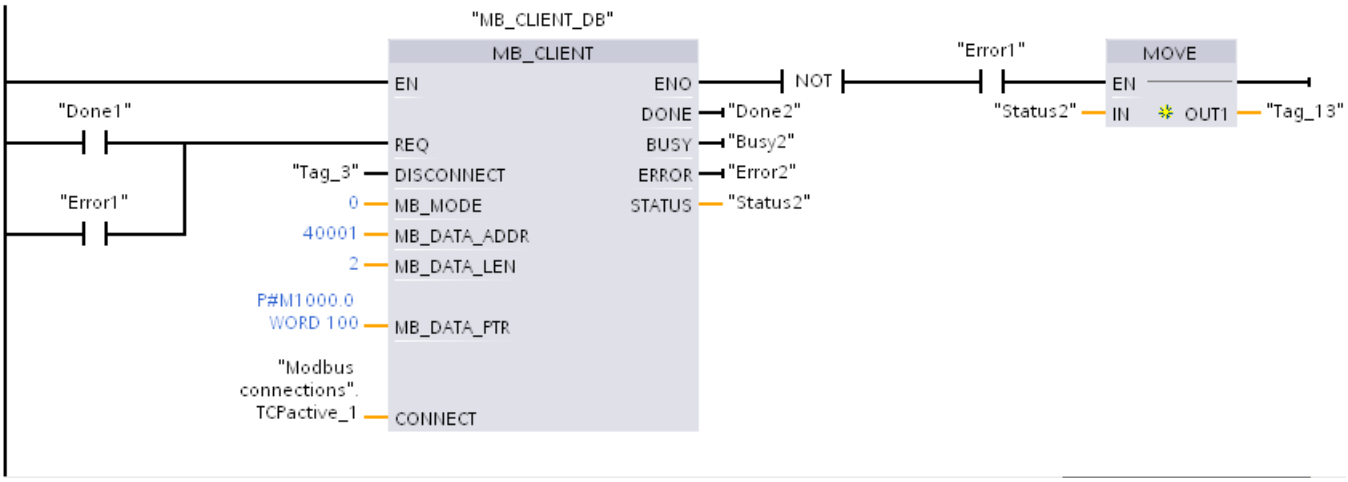
Необходимо обеспечить, чтобы выполнение каждого отдельного запроса Modbus TCP было бы завершено. Последовательностью выполнения должна управлять программная логика. Приведенный ниже пример демонстрирует, как выходы первого и второго клиентских запросов могут управлять последовательностью выполнения.

Пример показывает обоих клиентов, использующих те же данные соединения CONNECT (в различные моменты времени). Клиенты передают данные регистра хранения из того же удаленного сервера Modbus TCP по тому же адресу в M памяти локального CPU. Кроме того, возвращенная ошибка регистрируется, что является необязательным.

Сегмент 1: Modbus функция 3 – Считать слова регистра хранения на сервере Modbus TCP



Сегмент 2: Modbus функция 3 – Считать слова регистра хранения на сервере Modbus TCP



13.5.3 Modbus RTU

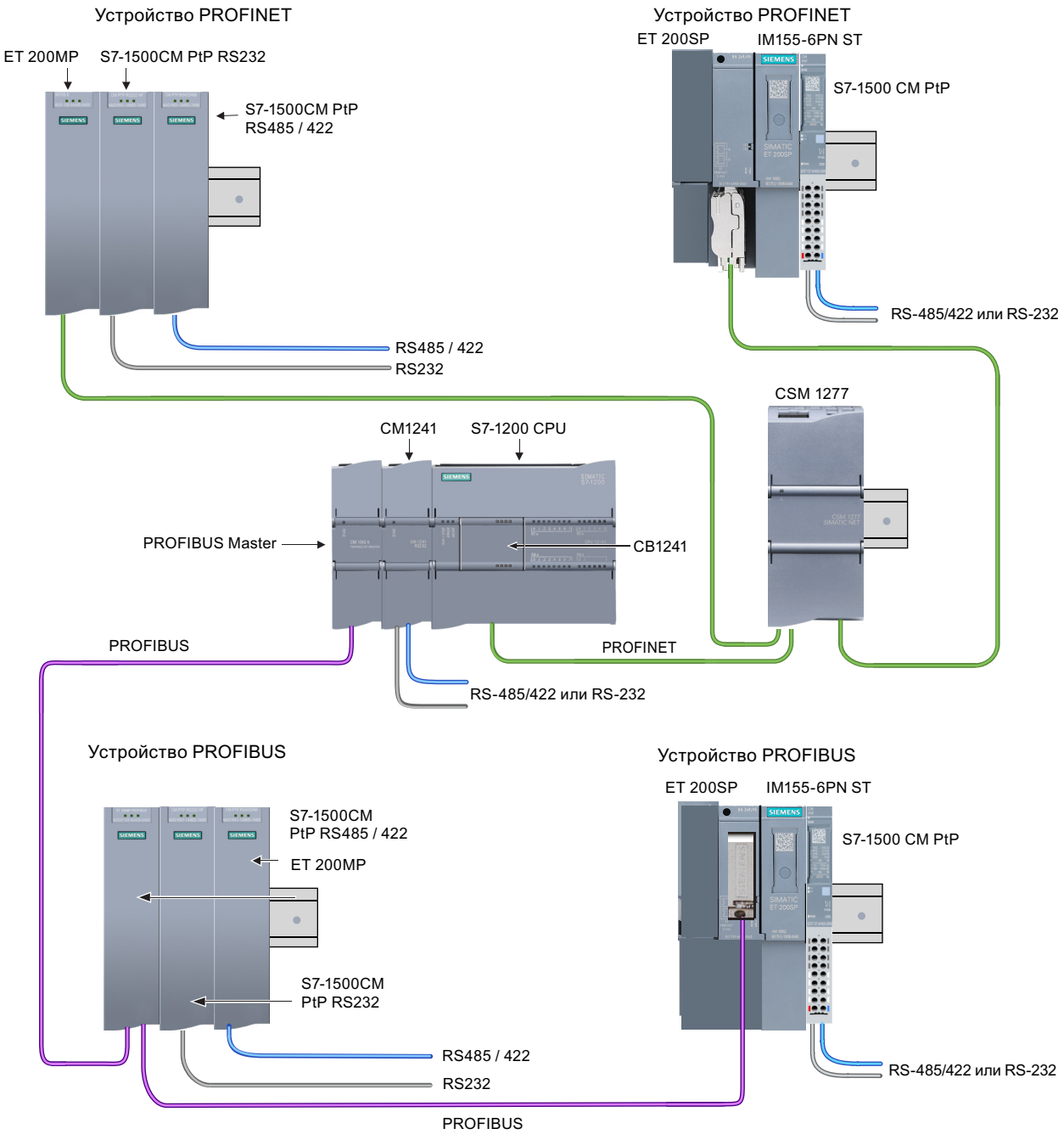
13.5.3.1 Обзор

От версии V4.1 S7-1200 CPU в комбинации с STEP 7 V13 SP1 расширяет возможности Modbus RTU для CPU для использования стойки распределенного ввода-вывода PROFINET или PROFIBUS для связи с различными устройствами (RFID-считыватели, устройство GPS и другие):

- PROFINET (Страница 810): Интерфейс Ethernet S7-1200 CPU подключается к интерфейсному модулю PROFINET. Коммуникационные модули PtP в стойке с интерфейсным модулем могут в этом случае обеспечить последовательную коммуникацию с PtP-устройствами.
- PROFIBUS (Страница 984): Коммуникационный модуль PROFIBUS устанавливается на левой стороне стойки с S7-1200 CPU. Коммуникационный модуль PROFIBUS соединяется со стойкой, содержащей интерфейсный модуль PROFIBUS. Коммуникационные модули PtP в стойке с интерфейсным модулем могут в этом случае обеспечить последовательную коммуникацию с PtP-устройствами.

Поэтому S7-1200 поддерживает два набора PtP-инструкций:

- Старые инструкции Modbus RTU (Страница 1347): Эти Modbus RTU инструкции использовались до версии V4.0 S7-1200 и обеспечивают последовательную коммуникацию только с использованием коммуникационного модуля CM 1241 или коммуникационной платы CB 1241.
- Инструкции Modbus RTU (Страница 1270): Эти Modbus RTU инструкции обеспечивают всю функциональность старых инструкций плюс возможность подключения к распределенному вводу-выводу PROFINET и PROFIBUS. Эти Modbus RTU инструкции позволяют конфигурировать связи между PtP коммуникационными модулями в стойке с распределенным вводом-выводом и PtP-устройствами. Для использования этих Modbus RTU инструкций S7-1200 модулям CM 1241 потребуется как минимум прошивка версии V2.1.



Примечание

От версии V4.1 S7-1200 инструкции PtP могут использоваться для любых типов коммуникации "точка-точка": последовательной, последовательной по PROFINET и последовательной по PROFIBUS. STEP 7 предлагает старые PtP инструкции только для поддержания имеющихся программ. Но старые инструкции работают и с CPU версии V4.1, а также с CPU версии 4.0 и более ранних версий. Преобразование старых программ под новые инструкции не требуется.

13.5.3.2 Выбор версии инструкций Modbus RTU

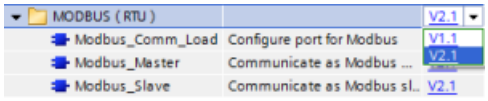
Следующие версии инструкций Modbus RTU доступны в STEP 7:

- Версия 3.4: совместима с CPU от версии V4.0 и CM от версии V2.1
- Версия 4.3: совместима с CPU от версии V4.2 и CM от версии V2.1

Учитывая совместимость и для упрощения миграции можно выбирать, какая версия инструкций будет вставлена в программу пользователя. Дополнительно S7-1200 CPU продолжает поддерживать старые версии инструкций Modbus RTU (Страница 1347).

Нельзя использовать обе версии инструкций для одного модуля, но два различных модуля могут использовать различные версии инструкций.

Вызвать на карте задач "Инструкция" инструкции MODBUS (RTU) в группе Коммуникационный процессор.



Для изменения версии инструкции Modbus RTU, выбрать соответствующую версию в раскрывающемся списке. Можно выбрать группу или отдельные инструкции.

Если инструкция Modbus RTU размещается через дерево инструкций в программе, то в дереве проекта создается новый экземпляр FB. Для просмотра нового экземпляра FB открыть в дереве проекта PLC_x > Программные блоки > Системные блоки > Программные ресурсы.

Чтобы узнать версию инструкции Modbus RTU в программе, следует вызвать свойства в дереве проекта, а не свойства блочного элемента на экране в редакторе текстов программ. Выбрать в дереве проекта экземпляр Modbus RTU FB, кликнуть правой кнопкой мыши, выбрать "Свойства" и после страницу "Информация", чтобы отобразить номер версии инструкции Modbus RTU.

13.5.3.3 Максимальное число поддерживаемых ведомых устройств Modbus

Адресация Modbus поддерживает максимум 247 ведомых устройств (номера ведомых устройств от 1 до 247). Каждый сегмент в сети Modbus может иметь до 32 устройств, в зависимости от возможностей загрузки и управления интерфейса RS485. По достижении ограничения в 32 устройства, следует использовать повторитель, чтобы перейти к следующему сегменту. Потребуется семь повторителей для поддержки 247 ведомых устройств, подключенных к одному ведущему устройству для RS485.

Повторители Siemens работают только с PROFIBUS. Их функция заключается в контроле прохождения токена PROFIBUS. Нельзя использовать повторители Siemens с другими протоколами. Поэтому для Modbus потребуются сторонние повторители.

Тайм-ауты Modbus по умолчанию длинные. Поэтому использование повторителей не создает проблем с задержкой. Работа ведущего устройства Modbus не зависит от того,

медленно ли реагирует ведомое устройство, или задерживают ли несколько ретрансляторов ответ.

13.5.3.4 Инструкции Modbus RTU

Инструкция Modbus_Comm_Load (конфигурирование SIPLUS I/O или порта на PtP модуле для Modbus RTU)

Таблица 13-75 Инструкция Modbus_Comm_Load

LAD/FBD	SCL	Описание

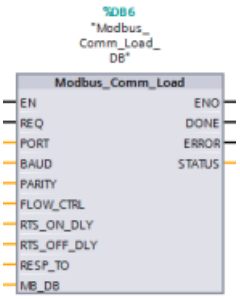
	`"Modbus_Comm_Load_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_uint_in_,`
	`BAUD:=_udint_in_,`
	`PARITY:=_uint_in_,`
	`FLOW_CTRL:=_uint_in_,`
	`RTS_ON_DLY:=_uint_in_,`
	`RTS_OFF_DLY:=_uint_in_,`
	`RESP_TO:=_uint_in_,`
	`DONE=>_bool_out,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`MB_DB:=_fbtref_inout_);`
		Инструкция Modbus_Comm_Load выполняет конфигурацию SIPLUS I/O или PtP-порта для коммуникации посредством протокола Modbus-RTU.
		Аппаратные опции порта Modbus RTU: Установка до трех CM (RS485 или RS232) плюс одна CB (R4845).
		Опции для Modbus RTU SIPLUS I/O: Установка ET 200MP S7-1500CM PtP (RS485 / 422 или RS232) или ET 200SP S7-1500 CM PtP (RS485 / 422 или RS232)
		При добавлении инструкции Modbus_Comm_Load в программу, автоматически назначается экземплярный блок данных.

Таблица 13-76 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
EN	IN	Bool	Примечание: Инструкция Modbus_Comm_Load для Modbus RTU использует инструкции PDREC и WRREC для инициализации PTP модуля. Но инструкция RDREC/WRREC является асинхронной, т.е. для ее выполнения потребуется несколько циклов. Поэтому параметр EN из Modbus_Comm_Load должен оставаться true до завершения инструкции RDREC/WRREC.
REQ	IN	Bool	Инструкцию запускает положительный фронт (0 на 1). (только версия 2.0)
PORT	IN	Port	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC.
BAUD	IN	UDInt	Выбор скорости передачи данных: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200. Все остальные значения недопустимы.

Параметр и тип данных		Тип данных	Описание
PARITY	IN	UInt	Выбор четности: • 0 – отсутствует • 1 – совпадение при контроле по нечётности • 2 – совпадение при контроле по чётности
FLOW_CTRL [1]	IN	UInt	Выбор управления потоком: • 0 – (по умолчанию) без управления потоком • 1 – аппаратное управления потоком с "RTS всегда ВКЛ" (не действует для портов RS485) • 2 – аппаратное управления потоком с "RTS с переключением"
RTS_ON_DLY [1]	IN	UInt	Выбор задержки включения RTS: • 0 – (по умолчанию) без задержки от активного RTS до передачи первого символа сообщения • От 1 до 65535 – Задержка в миллисекундах от активного RTS до передачи первого символа сообщения (не применимо к портам RS485). Задержки RTS должны применяться независимо от выбора для FLOW_CTRL.
RTS_OFF_DLY [1]	IN	UInt	Выбор задержки выключения RTS: • 0 – (по умолчанию) без задержки после передачи последнего символа и до "RTS неактивен" • От 1 до 65535 – задержка в миллисекундах от передачи последнего символа до "RTS неактивен" (не действует для портов RS485). Задержки RTS должны применяться независимо от выбора для FLOW_CTRL.
RESP_TO [1]	IN	UInt	Тайм-аут ответа: Время в миллисекундах, в течение которого Modbus_Master ждет ответа от ведомого устройства. Если ведомое устройство не отвечает в течение этого времени, Modbus_Master повторяет запрос или завершает запрос с ошибкой, если выполнено заданное количество повторов. От 5 до 65 535 мс (значение по умолчанию = 1000 мс).
MB_DB	IN	Variant	Ссылка на экземплярный блок данных инструкций Modbus_Master или Modbus_Slave. После вставки Modbus_Master или Modbus_Slave в программу, идентификатор DB будет доступен в выпадающем списке параметра на соединении блока MB_DB.
DONE	OUT	Bool	Бит DONE = TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок. (только версия 2.0)
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение TRUE, после того, как последний запрос был завершен с ошибкой. Код ошибки в параметре STATUS действителен только в цикле, где ERROR = TRUE.
STATUS	OUT	Word	Условие выполнения

[1] Опциональные параметры дляModbus_Comm_Load (V 2.x или выше). Кликнуть по стрелке у основания блока LAD/FBD, чтобы развернуть его и отобразить эти параметры.

Modbus_Comm_Load выполняется, чтобы выполнить конфигурацию порта для протокола Modbus-RTU. После того как была выполнена конфигурация порта для протокола Modbus RTU, он может быть использован инструкциями Modbus_Master или Modbus_Slave.

Для конфигурации каждого коммуникационного порта, который используется для коммуникации Modbus, необходимо исполнить Modbus_Comm_Load. Каждому используемому порту необходимо присвоить уникальный экземплярный блок данных Modbus_Comm_Load. Можно установить до трех коммуникационных модулей (RS232 или RS485) и одну коммуникационную плату (RS485) в CPU. Можно вызвать Modbus_Comm_Load из пускового OB и выполнить ее однократно или можно использовать системный флаг первого цикла (Страница 109), чтобы инициировать однократное выполнение. Повторное выполнение Modbus_Comm_Load требуется только тогда, если необходимо изменить параметры передачи данных, например скорость передачи данных или четность.

Если используется библиотека Modbus с модулем в распределенной стойке, инструкция Modbus_Comm_Load должна быть выполнена в процедуре циклического прерывания (например, один раз в секунду или один раз в 10 секунд). В случае прерывания питания распределенной стойки или извлечения модуля, после восстановления его работы только набор параметров HWConfig передается на PtP модуль. Все запросы, инициированные тайм-аутомModbus_Master, заканчиваются, и Modbus_Slave переходит в режим молчания (отсутствие ответа на любое сообщение). Циклическое выполнение инструкции Modbus_Comm_Load решает эти вопросы.

При добавлении Modbus_Master или Modbus_Slave в программу, инструкции назначается экземплярный блок данных. На этот экземплярный блок данных делается ссылка при указании параметра MB_DB в инструкции Modbus_Comm_Load.

Переменные (теги) блока данных экземпляра (DB) Modbus_Comm_Load

В таблице ниже приведены общедоступные статические переменные (теги) в экземплярном блоке данных Modbus_Comm_Load, которые могут быть использованы в программе.

Таблица 13-77 Статические переменные (теги) в экземплярном блоке данных Modbus_Comm_Load

Переменная (тег)	Тип данных	По умолчанию	Описание
ICHAR_GAP	Word	0	Макс. время задержки символа между символами. Данный параметр указывается в миллисекундах и предназначен для увеличения ожидаемого периода времени между принимаемыми символами. Соответствующее количество битовых интервалов для этого параметра добавляется к стандартному значению Modbus 35 битовых интервалов (3,5 x время символа).
RETRIES	Word	2	Количество повторных попыток, которые выполняет ведущее устройство, прежде чем будет выполнен возврат кода ошибки 0x80C8 для "ответ отсутствует".
EN_SUPPLY_VOLT	Bool	0	Активировать диагностику для отсутствующего напряжения питания L +.

Переменная (тег)	Тип данных	По умолчанию	Описание
MODE	USInt	0	Режим работы Возможны следующие режимы работы: • 0 = дуплекс (RS232) • 1 = дуплекс (RS422), 4-проводный режим (точка-точка) • 2 = дуплекс (RS422), 4-проводный режим (многоточечное ведущее устройство, CM PtP (ET 200SP)) • 3 = дуплекс (RS422), 4-проводный режим (многоточечное ведомое устройство, CM PtP (ET 200SP)) • 4 = полудуплекс (RS485), двухпроводный режим (см. примечание ниже)
LINE_PRE	USInt	0	Начальное состояние линии приема Возможны следующие начальные состояния: • 0 = начальное состояние "нет" (см. примечание ниже) • 1 = сигнал R(A) = 5 В DC, сигнал R(B) = 0 В DC (обнаружение обрыва провода): Обнаружение обрыва провода возможно в этом начальном состоянии. Выбор возможен только для: "Дуплекс (RS422) четырехпроводный режим (соединение точка-точка)" и "дуплекс (RS422) четырехпроводный режим (многоточечное ведомое устройство)". • 2 = сигнал R(A) = 0 В DC, сигнал R(B) = 5 В DC: Эта установка по умолчанию соответствует состоянию покоя (нет активных процессов передачи). Обнаружение обрыва провода в этом начальном состоянии невозможно.
BRK_DET	USInt	0	Обнаружение обрыва провода Возможны следующие настройки: • 0 = обнаружение обрыва провода выключено • 1 = обнаружение обрыва провода включено
EN_DIAG_ALARM	Bool	0	Активировать диагностическое прерывание: • 0 = не активировано • 1 = активировано
STOP_BITS	USInt	1	Количество стоповых битов: • 1 = 1 стоповый бит • 2 = 2 стоповых бита • 0, 3 до 255 = зарезервировано

Примечание

Необходимая установка для использования кабелей PROFIBUS с CM 1241 для RS485

Таблица 13-78 Коды состояния выполнения Modbus_Comm_Load [1]

STATUS (W#16#....)	Описание
0000	Ошибки отсутствуют
8180	Недопустимое значение ID порта (неправильный идентификатор порта/оборудования у коммуникационного модуля)
8181	Недопустимое значение для скорости передачи данных.
8182	Недопустимое значение для четности.
8183	Недопустимое значение для управления потоком.
8184	Недопустимое значение для тайм-аута ответа (тайм-аут ответа меньше, чем минимальное значение в 5 мс)
8185	Параметр MB_DB не является экземплярным блоком данных инструкции Modbus_Master или Modbus_Slave.

[1] Наряду с перечисленными выше ошибками инструкций Modbus_Comm_Load, базовые PtP коммуникационные инструкции также могут возвращать ошибки.

Инструкция Modbus_Master (обмен данными через SIPLUS I/O или PtP-порт в качестве ведущего устройства Modbus RTU)

Таблица 13-79 Инструкция Modbus_Master

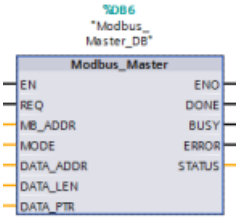
LAD/FBD	SCL	Описание
	**"Modbus_Master_DB"(** **REQ:=_bool_in_,** **MB_ADDR:=_uint_in_,** **MODE:=_usint_in_,** **DATA_ADDR:=_udint_in_,** **DATA_LEN:=_uint_in_,** **DONE=>_bool_out_,** **BUSY=>_bool_out_,** **ERROR=>_bool_out_,** **STATUS=>_word_out_,** **DATA_PTR:=_variant_inout_);**	Инструкция Modbus_Master выполняет коммуникацию как ведущее устройство Modbus через порт, конфигурация которого была при прошлом вызове инструкции Modbus_Comm_Load. При добавлении инструкции Modbus_Master в программу, автоматически назначается экземплярный блок данных. Этот экземплярный блок данных Modbus_Master используется при указании параметра MB_DB в инструкции Modbus_Comm_Load.

Таблица 13-80 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	0 = запрос отсутствует 1 = запрос для отправки данных на ведомое устройство Modbus
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Адрес станции Modbus-RTU: Стандартный диапазон адресации (от 1 до 247) Расширенный диапазон адресации (от 1 до 65535) Значение "0" зарезервировано для рассылки сообщения на все ведомые устройства Modbus. Только коды функций Modbus 05, 06, 15 и 16 поддерживаются широковещательной передачей.

Параметр и тип данных		Тип данных	Описание
MODE	IN	USInt	Выбор режима: Определяет тип запроса (чтение, запись или диагностика). Дополнительную информацию можно найти в таблице функций Modbus ниже.
DATA_ADDR	IN	UDInt	Начальный адрес в ведомом устройстве: Задает начальный адрес данных, к которым будет выполняться доступ в ведомом устройстве Modbus. Действительные адреса можно найти в таблице функций Modbus ниже.
DATA_LEN	IN	UInt	Длина данных: Указывает количество битов или слов, к которым будет обращаться этот запрос. Действительные значения длины можно найти в таблице функций Modbus ниже.
DATA_PTR	IN_OUT	Variant	Указатель данных: Указывает на M или DB адрес (не оптимизированный DB) для записываемых или считываемых данных.
DONE	OUT	Bool	Бит DONE = TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок.
BUSY	OUT	Bool	• 0 - Нет выполняемых операций Modbus_Master • 1 – Modbus_Master операция выполняется
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение TRUE, после того, как последний запрос был завершен с ошибкой. Код ошибки в параметре STATUS действителен только в цикле, где ERROR = TRUE.
STATUS	OUT	Word	Условие выполнения

Правила для коммуникации Modbus_Master

- Modbus_Comm_Load должна выполняться для конфигурирования порта, чтобы инструкция Modbus_Master могла связываться с этим портом.
- Если порт должен использоваться, чтобы инициировать запросы ведущего устройства Modbus, то порт не должен использоваться Modbus_Slave. Один или несколько экземпляров Modbus_Master могут использоваться с этим портом, но любое выполнение Modbus_Master должно использовать один и тот же экземплярный DB Modbus_Master для этого порта.
- Инструкции Modbus не используют коммуникационные прерывающие события для управления процессом коммуникации. Программа должна опрашивать инструкцию Modbus_Master на наличие завершенных процессов отправки и получения.

- Следует вызывать любое выполнение Modbus_Master для данного порта из OB программного цикла. Инструкции Modbus_Master могут выполняться только в одном из программных циклов или циклических/с временной задержкой уровнях выполнения. Они не должны выполняться на обоих уровнях приоритета выполнения. Приоритетное прерывание инструкции Modbus_Master другой инструкцией Modbus_Master на более высоком уровне приоритета выполнения приведет к нештатной работе. Инструкции Modbus_Master не должны обрабатываться на уровнях выполнения, связанных с запуском, диагностикой или ошибкой времени выполнения.
- Как только инструкция Modbus_Master инициирует передачу, этот экземпляр должен постоянно выполняться с активным входом EN, до возврата состояния DONE=1 или ERROR=1 . Определенный экземпляр Modbus_Master считают активным, пока не наступает одно из этих двух событий. В то время как исходный экземпляр активен, любой вызов другого экземпляра с активным входом REQ приведет к ошибке. Если непрерывное выполнение исходного экземпляра останавливается, состояние запроса остается активным на время, определенное статической переменной "Blocked_Proc_Timeout". Как только этот время истекает, следующая вызванная инструкция Modbus_Master с активным входом REQ станет активным экземпляром. Это препятствует тому, чтобы единственный экземпляр Modbus_Master монополизировал или заблокировал доступ к порту. Если исходный активный экземпляр не будет активирован в течение времени, определенного статической переменной "Blocked_Proc_Timeout", то следующее выполнение этого экземпляра (без установленного входа REQ) сбросит активное состояние. Если REQ установлен, то это выполнение инициирует новый запрос Modbus_Master, как будто никакой другой экземпляр не был активен.

Параметр REQ

0 = запрос отсутствует1 = запрос для отправки данных на ведомое устройство Modbus

Можно управлять этим входом с помощью активируемого уровнем или фронтом контакта. Каждый раз, когда этот вход активирован, запускается конечный автомат, чтобы гарантировать, что другая инструкция Modbus_Master, использующий тот же экземплярный DB, не выставить запрос, пока текущий запрос не завершен. Состояния всех других входов регистрируются и сохраняются в системе для текущего запроса, пока не будет получен ответ или не будет обнаружена ошибка.

Если тот же экземпляр Modbus_Master будет выполняться снова с входом REQ =1 до завершения текущего запроса, то последующие передачи не будут выполнены. Однако, когда запрос завершен, новый запрос выставляется каждый раз, когда Modbus_Master выполняется вновь с входом REQ = 1.

Через параметры DATA_ADDR и MODE выбирается тип функции Modbus.

DATA_ADDR (начальный адрес Modbus в ведомом устройстве): Задает начальный адрес данных, к которым будет выполняться доступ в ведомом устройстве Modbus.

Инструкция Modbus_Master использует вместо входа кода функции вход MODE. Комбинация из MODE и адреса Modbus определяет код функции, используемый в

собственном сообщении Modbus. В нижеследующей таблице показано отношение между параметром MODE, кодом функции Modbus и диапазоном адресов Modbus.

Таблица 13-81 Функции Modbus

MODE	Функция Modbus	Длина данных	Операция и данные	Адрес Modbus
0	01	От 1 до 2000 От 1 до 1992 [1]	Чтение выходных битов: От 1 до (1992 или 2000) бит за запрос	От 1 до 9999
0	02	От 1 до 2000 От 1 до 1992 [1]	Чтение входных битов: От 1 до (1992 или 2000) бит за запрос	От 10001 до 19999
0	03	От 1 до 125 От 1 до 124 [1]	Чтение регистра хранения: От 1 до (124 или 125) слов за запрос	От 40001 до 49999 или от 400001 до 465535
0	04	От 1 до 125 От 1 до 124 [1]	Чтение входных слов: От 1 до (124 или 125) слов за запрос	От 30001 до 39999
104	04	От 1 до 125 От 1 до 124 1	Чтение входных слов: От 1 до (124 или 125) слов за запрос	От 00000 до 65535
1	05	1	Запись выходного бита: Один бит на запрос	От 1 до 9999
1	06	1	Запись регистра хранения: 1 слово на запрос	От 40001 до 49999 или от 400001 до 465535
1	15	От 2 до 1968 От 2 до 1960 [1]	Запись нескольких выходных битов: От 2 до (1960 или 1968) бит за запрос	От 1 до 9999
1	16	От 2 до 123 От 2 до 122 [1]	Запись нескольких регистров хранения: От 2 до (122 или 123) слов за запрос	От 40001 до 49999 или от 400001 до 465535
2	15	От 1 до 1968 От 2 до 1960 [1]	Запись одного или нескольких выходных битов: От 1 до (1960 или 1968) бит за запрос	От 1 до 9999
2	16	От 1 до 123 От 1 до 122 [1]	Запись одного или нескольких регистров хранения: От 1 до (122 или 123) слов за запрос	От 40001 до 49999 или от 400001 до 465535
11	11	0	Считать слово состояния и счетчик событий коммуникации ведомого устройства. Статусное слово указывает на "занято" (0 – не занято, 0xFFFF – занято). Счетчик событий инкрементируется при каждой удачной обработке сообщения. Оба операнда DATA_ADDR и DATA_LEN из Modbus_Master игнорируются данной функцией.	
80	08	1	Проверка состояния ведомого устройства диагностическим кодом данных 0x0000 (петлевой тест – ведомое устройство возвращает эхо запроса) 1 слово на запрос	

MODE	Функция Modbus	Длина данных	Операция и данные	Адрес Modbus
81	08	1	Сброс счетчика событий ведомого устройства посредством диагностического кода данных 0x000A 1 слово на запрос	
3 до 10, 12 до 79, 82 до 255			Зарезервировано	

[1] При расширенной адресации максимальная длина данных, в зависимости от типа данных функции, меньше на 1 байт или 1 слово.

Параметр DATA_PTR

Параметр DATA_PTR указывает на адрес в DB или M области, в который выполняется записи или из которого выполняется чтение. При использовании блока данных необходимо создать глобальный блок данных, который предоставит память данных для операций чтения и записи на ведомые устройства Modbus.

Примечание

Блок данных DATA_PTR должен допускать прямую адресацию.

Блок данных должен допускать прямую (абсолютную) и символьную адресацию. При создании блока данных следует выбрать атрибут доступа "Стандарт".

Начиная с версии 4.0 инструкции Modbus_Master, можно активировать атрибут блока данных "Оптимизированный доступ к блоку". Можно использовать только один элемент или массив элементов в оптимизированной памяти со следующими типами данных: Bool, Byte, Char, Word, Int, DWord, Dint, Real, USInt, UInt, UDInt, SInt или WChar.

Структуры блока данных для параметра DATA_PTR

- Эти типы данных действуют для **считывания слов** адресов Modbus от 30001 до 39999, от 40001 до 49999 и от 400001 до 465536, а также для **записи слов** в адреса Modbus от 40001 до 49999 и от 400001 до 465536.
  - Стандартный массив типов данных WORD, UINT или INT.
  - Именованная структура типа WORD, UINT или INT, где каждый элемент имеет уникальное имя и 16-битный тип данных.
  - Именованная сложная структура, где каждый элемент имеет уникальное имя и 16- или 32-битный тип данных.
- Для **чтения** и записи битов для адресов Modbus с 00001 по 09999 и для чтения битов с 10001 по 19999.
  - Стандартный массив булевых типов данных.
  - Именованная булева структура уникально именованных булевых переменных.

- Наличие отдельной области памяти для каждой инструкции Modbus_Master не обязательно, но рекомендуется. Причина этой рекомендации кроется в том, что есть большая вероятность повреждения данных, если несколько инструкций Modbus_Master читают и пишут в ту же самую область памяти.
- Не обязательно, чтобы области данных для DATA_PTR находились в одном и том же глобальном блоке данных. Можно создать один блок данных с несколькими областями для процессов считывания Modbus, один блок данных для процессов записи Modbus или один блок данных для каждой станции ведомого устройства.

Переменные в блоке данных (DB) инструкции Modbus_Master

В таблице ниже приведены общедоступные статические переменные (теги) в экземплярном блоке данных Modbus_Master, которые могут быть использованы в программе.

Таблица 13-82 Статические переменные в экземплярном блоке данных Modbus_Master

Переменная (тег)	Тип данных	По умолчанию	Описание
Blocked_Proc_Timeout	Real	3.0	Время (в секундах), в течение которого ожидается заблокированный экземпляр Modbus_Master, прежде чем этот экземпляр будет удален со статусом ACTIVE. Такое, например, возможно, если выдается запрос Modbus_Master и программа прекращает вызов функции вModbus_Master, прежде чем запрос будет полностью завершен. Значение времени должно быть больше 0 и меньше 55 секунд. В противном случае появится ошибка.
Extended_Addressing	Bool	FALSE	Конфигурирует одно- или двухбайтовую адресацию ведомой станции: • FALSE = однобайтный адрес; от 0 до 247 • TRUE = двухбайтный адрес (соответствует расширенной адресации); от 0 до 65535
MB_DB	MB_BASE	-	Параметр MB_DB инструкции Modbus_Comm_Load должен быть соединен с параметром MB_DB инструкции Modbus_Master.

Программа может записывать значения в переменные Blocked_Proc_Timeout и Extended_Addressing , чтобы управлять операциями Modbus_Master. В разделе Modbus_Slave в описании HR_Start_Offset (Страница 1282) и Extended_Addressing (Страница 1282) можно найти пример того, как использовать эти переменные в редакторе текстов программ, а также другая подробная информация о расширенной адресации Modbus.

Коды условий

Таблица 13-83 Коды условий для выполнения Modbus_Master (ошибки коммуникации и конфигурации) [1]

STATUS (W#16#....)	Описание
0000	Ошибки отсутствуют
80C8	Тайм-аут ведомого устройства. Указанное ведомое устройство не ответило в течение определенного времени. Проверить скорость передачи данных, четность и подключения ведомого устройства. Об этой ошибке сообщается только после того, как были предприняты все сконфигурированные повторы.
80C9	Время инструкции Modbus_Master истекло по одной из следующих причин: • Инструкция ожидает ответа от модуля, который используется для коммуникаций. • Установлено слишком маленькое значение Blocked_Proc_Timeout. Об этой ошибке сообщается, если PROFIBUS- или PROFINET-устройство распределенного ввода-вывода восстановилось после одного из следующих событий: • Прерывание питания или коммуникаций • Событие извлечения/вставки коммуникационного модуля В этих случаях перезагружается аппаратная конфигурация из PLC , и инструкция Modbus_Comm_Load должна быть выполнен снова, чтобы должным образом сконфигурировать коммуникационный модуль.
80D1	Получатель запросил управление потоком для прерывания текущей передачи, но не активировал передачу снова в течение времени ожидания. Эта ошибка также генерируется при аппаратном управлении потоком, если получатель в течение времени ожидания не распознает CTS.
80D2	Запрос на передачу был прерван, так как сигнал DSR не получен от DCE.
80E0	Сообщение завершено, так как принимающий буфер заполнен.
80E1	Сообщение завершено из-за ошибки четности.
80E2	Сообщение завершено из-за ошибки телеграммы.
80E3	Сообщение завершено из-за ошибки переполнения.
80E4	Сообщение отменено, так как указанная длина превышает суммарный размер буфера.
8180	Недопустимое значение для ID порта или ошибка инструкции Modbus_Comm_Load
8186	Недействительный адрес станции Modbus
8188	Недопустимый режим определен для широковещательного запроса
8189	Недопустимое значение для адреса данных.
818A	Недопустимое значение для длины данных.
818B	Недействительный указатель на локальный источник данных, т. е. на локальную цель данных: Неверный размер
818C	Недействительный указатель для DATA_PTR или Blocked_Proc_Timeout. Область данных должна быть одной из следующих: • Классический DB • Массив элементарных типов данных в символьном или энергонезависимом DB • M память
8200	Порт занят обработкой запроса передачи.

STATUS (W#16#....)	Описание
8280	Отрицательное подтверждение при чтении из модуля Проверить вход в параметре PORT. Эта ошибка может быть вызвана потерей PROFIBUS, или PROFINET модуля распределенного ввода-вывода, потерей питания или обмена данными, либо извлечением модуля.
8281	Отрицательное подтверждение при записи в модуль Проверить вход в параметре PORT. Эта ошибка может быть вызвана потерей PROFIBUS, или PROFINET модуля распределенного ввода-вывода, потерей питания или обмена данными, либо извлечением модуля.

Таблица 13-84 Коды условий для выполнения Modbus_Master (ошибка протокола Modbus) [1]

STATUS (W#16#....)	Код ответа ведомого устройства	Ошибка протокола Modbus
8380	-	Ошибка контр. суммы
8381	01	Код функции не поддерживается
8382	03	Ошибка в длине данных
8383	02	Ошибка в адресе данных или адрес вне действительного диапазона DATA_PTR
8384	Больше чем 03	Ошибка в значении данных
8385	03	Значение диагностического кода данных не поддерживается (код функции 08)
8386	-	Код функции в ответе не соответствует коду в запросе.
8387	-	Ответило неправильное ведомое устройство.
8388	-	Ответ ведомого устройства на запрос записи неправильный. Запрос записи, возвращенный ведомым устройством, не соответствует отправленному ведущим устройством.

[1] Наряду с перечисленными выше ошибками инструкций Modbus_Master, базовые PtP коммуникационные инструкции также могут возвращать ошибки.

Примечание

Установка макс. длины записи для коммуникации Profibus

Если коммуникационный модуль CM1243-5 выступает в роли ведущего устройства Profibus для управления Profibus устройством ET200SP или ET200MP, которое в свою очередь используется модуль точка-точка RS232, RS422 или RS485, переменная (тег) блока данных "max_record_len" должна быть явно установлена на значение 240 следующим образом:

Установить после выполнения Modbus_Comm_Load параметр max_record_len в разделе Send_P2P DB экземпляра (напр., "Modbus_Master_DB".Send_P2P.max_record_len) на 240.

Явное присвоение значения для max_record_len необходимо только для Profibus соединений. Для Profinet соединений для max_record_len уже используется правильное значение.

Инструкция Modbus_Slave (обмен данными через SIPLUS I/O или PtP-порт в качестве ведомого устройства Modbus RTU)

Таблица 13-85 Инструкция Modbus_Slave

LAD/FBD	SCL	Описание

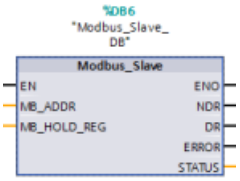
	`"Modbus_Slave_DB"(`
	`MB_ADDR:=_uint_in_,`
	`NDR=>_bool_out_,`
	`DR=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`MB_HOLD_REG:=_variant_inout_);`
		Инструкция Modbus_Slave позволяет программе выполнять обмен одним из двух способов:
		• В качестве ведомого устройства Modbus RTU через PtP порт на CM (RS485 или RS232) и CB (RS485)
		• В качестве ведомого устройства Modbus RTU через опции Modbus RTU SIPLUS I/O:
		– Установить ET 200MP S7-1500CM PtP (RS485 / 422 или RS232).
		– Установить ET 200SP S7-1500 CM PtP (RS485 / 422 или RS232).
		Когда удаленное ведущее устройство Modbus RTU выставляет запрос, пользовательская программа отвечает на запрос выполнением инструкции Modbus_Slave. При добавлении инструкции, STEP 7 автоматически создает DB экземпляра. С помощью этого имени Modbus_Slave_DB задается параметр MB_DB для инструкции Modbus_Comm_Load.

Таблица 13-86 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Адрес станции ведомого устройства Modbus: Стандартный диапазон адресации (от 1 до 247) Расширенный диапазон адресации (от 0 до 65535)
MB_HOLD_REG	IN_OUT	Variant	Указатель на DB регистра хранения Modbus: регистр хранения Modbus может быть M памятью или блоком данных.
NDR	OUT	Bool	Новые данные готовы: • 0 – Новых данных нет • 1 – Указывает, что новые данные были записаны ведущим устройством Modbus
DR	OUT	Bool	Считывание данных: • 0 – Данные не считаны • 1 – Указывает, что данные были считаны ведущим устройством Modbus

Параметр и тип данных		Тип данных	Описание
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение TRUE, после того, как последний запрос был завершен с ошибкой. Если исполнение завершается ошибкой, код ошибки в параметре STATUS действителен только на протяжении цикла, в котором ERROR = TRUE.
STATUS	OUT	Word	Код ошибки выполнения

Коды функций коммуникации Modbus (1, 2, 4, 5 и 15) могут считывать и записывать биты и слова непосредственно в образе процессов входов и в образе процессов выходов CPU. Для этих функциональных кодов параметр MB_HOLD_REG в качестве типа данных должен быть больше байта. Следующая таблица отражает примерное присвоение адресов Modbus для образа процесса в CPU.

Таблица 13-87 Согласование адресов Modbus для образа процесса

Функции Modbus						S7-1200	
Коды	Функция	Область данных	Диапазон адресов			Область данных	Адрес CPU
01	Читать биты	Вывод	1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7
02	Читать биты	Ввод	10001	до	18192	Образ процессов по вводу	От I0.0 до I1023.7
04	Чтение слов	Ввод	30001	до	30512	Образ процессов по вводу	От IW0 до IW1022
05	Записать бит	Вывод	1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7
15	Записать биты	Вывод	1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7

Функциональные коды коммуникации Modbus (3, 6, 16) используют регистр хранения, под которым может подразумеваться диапазон адресов в M памяти или блок данных. Тип регистра хранения указывает параметр MB_HOLD_REG инструкции Modbus_Slave.

Примечание

Тип блока данных MB_HOLD_REG

Блок данных с регистром хранения Modbus должен разрешать прямую (абсолютную) и символьную адресацию. При создании блока данных следует выбрать атрибут доступа "Стандарт".

Начиная с версии 4.0 инструкции Modbus_Slave, можно активировать атрибут блока данных "Оптимизированный доступ к блоку". Можно использовать только один элемент или массив элементов в оптимизированной памяти со следующими типами данных: Bool, Byte, Char, Word, Int, DWord, Dint, Real, USInt, UInt, UDInt, SInt или WChar.

Следующая таблица показывает примеры согласования адресов Modbus с регистрами хранения, используемые для кодов функций Modbus 03 (чтение слов), 06 (запись слов)

и 16 (запись слов). Фактический верхний предел адресов DB определен максимальным объемом рабочей памяти и объемом M памяти для каждой из моделей CPU.

Таблица 13-88 Согласование адресов Modbus для памяти CPU

Адрес ведущего устройства Modbus	**Примеры для параметра MB_HOLD_REG**				
	MW100	**DB10.DBW0**	**MW120**	**DB10.DBW50**	**"Recipe".ingredient**
40001	MW100	DB10.DBW0	MW120	DB10.DBW50	"Recipe".ingredient[1]
40002	MW102	DB10.DBW2	MW122	DB10.DBW52	"Recipe".ingredient[2]
40003	MW104	DB10.DBW4	MW124	DB10.DBW54	"Recipe".ingredient[3]
40004	MW106	DB10.DBW6	MW126	DB10.DBW56	"Recipe".ingredient[4]
40005	MW108	DB10.DBW8	MW128	DB10.DBW58	"Recipe".ingredient[5]

Таблица 13-89 Функции диагностики

Функции диагностики Modbus_Slave для S7-1200		
Коды	Подфункция	Описание
08	0000H	Возврат данных запроса эхо-теста: • До STEP 7 V15.1 инструкция Modbus_Slave возвращает ведущему устройству Modbus эхо полученного слова данных. • От STEP 7 V15.1 инструкция Modbus_Slave V4.1 или выше возвращает одно или несколько полученных слов данных.
08	000AH	Очистка счетчика событий коммуникации: Инструкция Modbus_Slave очищает счетчик событий коммуникации, который используется для функции Modbus 11.
11		Вызов счетчика событий коммуникации: Инструкция Modbus_Slave использует внутренний счетчик событий коммуникации, чтобы определять количество выполненных запросов чтения и записи Modbus, которые отправляются на Modbus_Slave. Значение счетчика не увеличивается при выполнении функций 8, функции 11 и запросов широкого вещания. Кроме этого, значение счетчика не увеличивается при запросах, которые приводят к ошибкам коммуникации (например, ошибка четности или контрольной суммы).

Инструкция Modbus_Slave поддерживает широковещательные запросы записи от любого ведущего устройства Modbus, если запрос использует допустимые адреса. Modbus_Slave создает код ошибки "0x8188" для кодов функций, не поддерживаемых в широковещательной передаче.

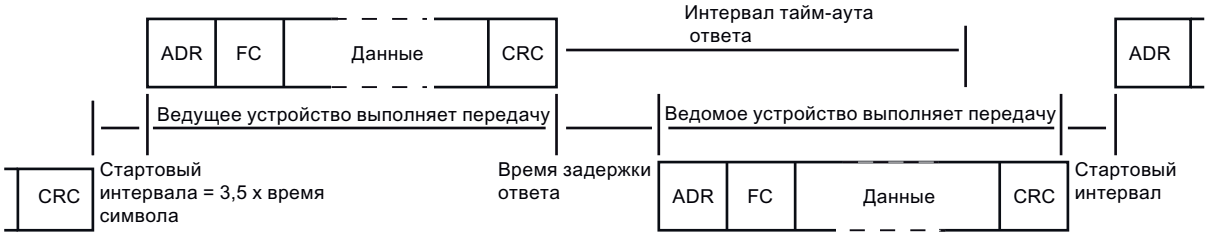
Правила для коммуникации Modbus_Slave

- Modbus_Comm_Load должна выполняться для конфигурирования порта, чтобы инструкция Modbus_Slave могла выполнить коммуникацию через этот порт.
- Если порт должен реагировать на Modbus_Master как ведомое устройство, программирование этого порта при помощи инструкции Modbus_Master запрещено.
- С определенным портом может использоваться только один экземпляр Modbus_Slave, в противном случае возможно неожиданное поведение.

- Инструкции Modbus не используют коммуникационные прерывающие события для управления процессом коммуникации. Программа должна считывать коммуникационный процесс путем опроса инструкции Modbus_Slave на наличие завершенных процессов отправки и получения.
- Инструкция Modbus_Slave должна выполняться регулярно с частотой, которая позволяет давать в указанное время ответ на входящие запросы Modbus_Master. Рекомендуется выполнять Modbus_Slave в каждом цикле из OB программного цикла. Modbus_Slave может выполняться из OB циклических прерываний, однако это не рекомендуется, так как из-за слишком больших задержек по времени в прерывающей программе, выполнение других программ прерывания может быть временно заблокировано.

Временная диаграмма сигнала Modbus

Modbus_Slave должна выполняться регулярно, чтобы принимать каждый запрос Modbus_Master и отвечать на него соответствующим образом. Частота выполнения Modbus_Slave зависит от значения тайм-аута ответа, заданного Modbus_Master. Это показано на следующем рисунке.



Период тайм-аута для ответа RESP_TO это промежуток времени, в течение которой Modbus_Master ожидает начала ответа от Modbus_Slave. Этот период времени определяется не протоколом Modbus, а параметром соответствующего Modbus_Master. Периодичность выполнения (время между текущим и последующим выполнениями) Modbus_Slave должна базироваться на определенных параметрах Modbus_Master. Инструкцию Modbus_Slave необходимо выполнять как минимум два раза во время периода тайм-аута для ответа Modbus_Master.

Переменные в блоке данных (DB) инструкции Modbus_Slave

В таблице ниже приведены общедоступные статические переменные (теги) в экземплярном блоке данных Modbus_Slave, которые могут быть использованы в программе.

Таблица 13-90 Статические переменные в экземплярном блоке данных Modbus_Slave

Переменная (тег)	Тип данных	По умолчанию	Описание
HR_Start_Offset	Word	0	Задает начальный адрес регистра хранения Modbus (по умолчанию = 0)
Extended_Addressing	Bool	FALSE	Выполняет конфигурацию адреса станции ведомого устройства в виде единичного или двойного байта: • FALSE = 1-байтовый адрес • TRUE = 2-байтовый адрес
Request_Count	Word	0	Общее количество запросов, полученных этим ведомым устройством
Slave_Message_Count	Word	0	Количество запросов, полученных для этого конкретного ведомого устройства
Bad_CRC_Count	Word	0	Количество полученных запросов, содержащих ошибку контрольной суммы
Broadcast_Count	Word	0	Количество полученных широковещательных запросов
Exception_Count	Word	0	Конкретные ошибки Modbus, требующие квитирования с возвращением исключения на ведущее устройство
Success_Count	Word	0	Количество полученных для этого конкретного ведомого устройства запросов без ошибок протокола
MB_DB	MB_BASE	-	Параметр MB_DB инструкции Modbus_Comm_Load должен быть соединен с параметром MB_DB инструкции Modbus_Slave.
QB_Start	UInt	0	Начальный адрес выходных байт, в которые может выполнять запись CPU (от AB0 до AB65535)
QB_Count	UInt	65535	Количество байтов, в которые может выполнять запись удаленное устройство. Если QB_Count = 0, то удаленное устройство не может записывать на выходы. Пример: Если запись должна быть возможна только в AB10 - AB17, то установка QB_Start = 10 и QB_Count = 8.
QB_Read_Start	UInt	0	Начальный адрес выходных байт, из которых может выполнять чтение CPU (от AB0 до AB65535)
QB_Read_Count	UInt	65535	Количество выходных байт, из которых может считывать удаленное устройство. Если QB_Count = 0, то удаленное устройство не может читать из выходов. Пример: Если чтение должно быть возможным только для AB10 - AB17, то установка QB_Start = 10 и QB_Count = 8.
IB_Read_Start	UInt	0	Начальный адрес входных байтов, из которых может выполнять чтение CPU (от EB0 до EB65535)
IB_Read_Count	UInt	65535	Количество входных байтов, из которых может считывать удаленное устройство. Если IB_Count = 0, то удаленное устройство не может читать из входов. Пример: Если чтение должно быть возможным только для EB10 - EB17, то установка IB_Start = 10 и IB_Count = 8.

Программа может записывать данные в операции управления сервера Modbus и следующие переменные:

- HR_Start_Offset
- Extended_Addressing
- QB_Start
- QB_Count
- QB_Read_Start
- QB_Read_Count
- IB_Read_Start
- IB_Read_Count

Для доступности переменных в блоке данных (DB) инструкции Modbus_Slave существуют следующие требования к версиям

Таблица 13-91 Требования к версиям для доступности переменных в блоке данных (DB) инструкции Modbus_Slave Инструкция, TIA Portal и S7-1200 CPU

Версия инструкции Modbus_Slave	**Версия TIA Portal**	**Версия прошивки S7-1200 CPU**	**Переменные блоков данных**
3.0	V14 SP1	CPU FW V4.0 или выше	QB_Start
			QB_Count
4.0 или выше	V15 или выше	CPU FW V4.2 или выше	QB_Start
			QB_Count
			QB_Read_Start
			QB_Read_Count
			IB_Read_Start
			IB_Read_Count

HR_Start_Offset

Адреса регистра хранения Modbus начинаются с 40001 или 400001. Эти адреса соответствуют начальному адресу регистра хранения в памяти целевой системы. Однако можно использовать переменную HR_Start_Offset, чтобы сконфигурировать начальный адрес для регистра хранения Modbus, отличный от 40001 или 400001.

Можно, например, сконфигурировать регистр хранения с началом в MW100 и длиной в 100 слов. При помощи смещения 20 можно задать начальный адрес регистра хранения 40021 вместо 40001. Любой адрес меньше 40021 и больше 400119 приводит к ошибке адресации.

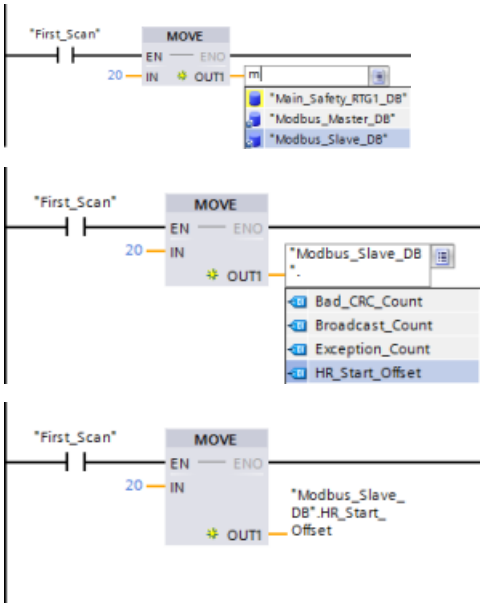
Таблица 13-92 Примеры для адресации регистра хранения Modbus

HR_Start_Offset	**Адрес**	**Минимум**	**Максимум**
0	Адрес Modbus (Word)	40001	40099
	Адрес S7-1200	MW100	MW298

HR_Start_Offset	Адрес	Минимум	Максимум
20	Адрес Modbus (Word)	40021	40119
	Адрес S7-1200	MW100	MW298

HR_Start_Offset - значение типа word, которое определяет начальный адрес регистра хранения Modbus и сохраняется в экземплярном блоке данных Modbus_Slave. Эти общедоступные статические переменные можно выбрать из выпадающего списка параметров, после того как в программу была вставлена инструкция Modbus_Slave.

Если, например, после вставки Modbus_Slave в сегмент LAD, можно перейти в предыдущий сегмент и назначить значение HR_Start_Offset. Значение необходимо назначить до исполнения Modbus_Slave.



Ввод переменной ведомого устройства Modbus через имя блока данных по умолчанию:

1. Установить курсор в поле параметра и ввести символ m.
2. В выпадающем списке выбрать "Modbus_Slave_DB".
3. Установить курсор справа от имени блока данных (после кавычек) и ввести точку.
4. В выпадающем списке выбрать "Modbus_Slave_DB.HR_Start_Offset".

Extended_Addressing

Доступ к переменной Extended_Addressing осуществляется аналогичным способом, как и к представленной выше ссылке HR_Start_Offset, за исключением того случая, когда переменная Extended_Addressing является булевым значением. Булева переменная должна быть записана с помощью выходной катушки, а не блока перемещения.

Адресация ведомого устройства Modbus может быть сконфигурирована либо как однобайтовая (что является стандартом Modbus), либо, как двухбайтовая. Расширенная адресация используется для адресации более 247 устройств в одной сети. Выбор

расширенной адресации позволяет адресовать максимум 64.000 адресов. В дальнейшем в качестве примера будет рассмотрена телеграмма функции 1 Modbus.

Таблица 13-93 Однобайтовый адрес ведомого устройства (байт 0)

Функция 1	**Байт 0**	**Байт 1**	**Байт 2**	**Байт 3**	**Байт 4**	**Байт 5**	
Запрос	Адрес ведомого устройства	Код функции	Начальный адрес		Длина катушек		
Действительный ответ	Адрес ведомого устройства	Код функции	Длина	Данные катушек			
Ответ с ошибкой	Адрес ведомого устройства	0x81	Код ошибки				

Таблица 13-94 Двухбайтовый адрес ведомого устройства (байты 0 и 1)

	Байт 0	**Байт 1**	**Байт 2**	**Байт 3**	**Байт 4**	**Байт 5**	**Байт 6**
Запрос	Адрес ведомого устройства		Код функции	Начальный адрес		Длина катушек	
Действительный ответ	Адрес ведомого устройства		Код функции	Длина	Данные катушек		
Ответ с ошибкой	Адрес ведомого устройства		0x81	Код ошибки			

Обращение к областям данных в блоках данных (DB) вместо прямого доступа к адресам Modbus

От версии V4.0 инструкции Modbus_Slave и версии V4.2 прошивки S7-1200 CPU возможно обращение к областям данных в DB, вместо прямого обращения к образам процесса и регистрам хранения. Для этого на странице свойств "Атрибуты" глобального DB необходимо снять флажки "Помещать только в загружаемую память" и "Оптимизированный доступ к блоку".

Если поступает запрос Modbus и область данных для типа данных Modbus соответствующего кода функции не была определена, то инструкция Modbus_Slave обрабатывает запрос согласно процедуре в прежних версиях инструкции: Можно обращаться напрямую к образам процесса и регистрам хранения.

Если область данных для типа данных Modbus кода функции была определена, то инструкция Modbus_Slave выполняет чтение из этой области данных и выполняет запись в эту область данных. Процесс чтение или записи зависит от типа задания.

Примечание

Если область данных сконфигурирована, то инструкция Modbus_Slave игнорирует смещения и области, сконфигурированные статическими переменными в блоке данных экземпляра, который соответствует типу данных в области данных. Эти смещения и области действительны только для образа процесса или памяти, на которые ссылается MB_HOLD_REG. Параметры начала и длины области данных предлагают собственный способ для определения смещений и областей.

Для индивидуального запроса Modbus возможна запись только в одну область данных или чтение из нее. Например, если требуется чтение регистров хранения, занимающих несколько областей данных, то потребуется несколько запросов Modbus.

Для определения областей данных действуют следующие правила:

- Можно определить до восьми областей данных в различных DB, при этом каждый DB может содержать только одну область данных. Для одного запроса Modbus возможна запись только в одну область данных или чтение из нее. Каждая область данных соответствует одному диапазону адресов MODBUS. Области данных устанавливаются в статической переменной "Data_Area_Array" экземплярного DB.
- Если необходимо использовать менее восьми областей данных, то следует разместить необходимые области данных одну за другой без пропусков. На первой пустой записи в областях данных поиск области данных во время обработки отменяется. Если, например, элементы поля 1, 2, 4 и 5 были определены как области данных, то "Data_Area_Array" распознает только элементы поля 1 и 2, поскольку элемент поля 3 пуст.

- После Data_Area_Array включает в себя восемь элементов: от Data_Area_Array[1] до Data_Area_Array[8]
- Каждый элемент поля Data_Area_Array[x], 1 <= x <= 8 - это UDT типа MB_DataArea со следующей структурой:

Параметр	Тип данных	Значение
data_ type	UInt	Идентификатор для типа данных MODBUS, который отображен на эту область данных: • 0: Идентификатор для пустого элемента поля или неиспользуемой области данных. В этом случае значения для db, start и length не имеют значения. • 1: Выход образа процесса (в комбинации с кодами функции 1, 5 и 15) • 2: Вход образа процесса (в комбинации с кодом функции 2) • 3: Регистр хранения (в комбинации с кодами функции 3, 6 и 16) • 4: Входной регистр (в комбинации с кодом функции 4) Примечание: Если для типа данных MODBUS была определена область данных, то инструкция Modbus_Slave больше не сможет напрямую обращаться к этому типу данных MODBUS. Если адрес запроса MODBUS для такого типа данных не соответствует установленной области данных, то в STATUS возвращается значение W#16#8383.
db	UInt	Номер блока данных, на который отображаются регистр MODBUS или определенные после биты Номер DB должен быть оригинальным в областях данных. Номер DB не может встречаться в нескольких областях данных. На странице свойств "Атрибуты" глобального DB необходимо снять флажки "Помещать только в загружаемую память" и "Оптимизированный доступ к блоку". Области данных также начинаются с адрес байта 0 в DB. Допустимые значения: от 1 до 60999
start	UInt	Первый адрес MODBUS, отображенный на блок данных, начинающийся с адреса 0.0 Допустимые значения: от 0 до 65535
length	UInt	Число бит (для значений 1 и 2 из data_type) или число регистров (для значений 3 и 4 из data_type) Диапазоны адресов MODBUS одного типа данных MODBUS не должны накладываться друг на друга. Допустимые значения: от 1 до 65535

Примеры для определения областей данных:

- Первый пример: data_type = 3, db = 1, start = 10, length = 6
  CPU отображает регистры хранения (data_type = 3) в блоке данных 1 (db = 1) и устанавливает Modbus адрес 10 (start = 10) на слово данных 0, а последний действительный Modbus адрес 15 (length = 6) на слово данных 5.
- Второй пример: data_type = 2, db = 15, start = 1700, length = 112
  CPU отображает входы (data_type = 2) в блоке данных 15 (db = 15) и устанавливает Modbus адрес 1700 (start = 1700) на слово данных 0, а последний действительный Modbus адрес 1811 (length = 112) на слово данных 111.

Коды условий

Таблица 13-95 Коды условий для выполнения Modbus_Slave (ошибки коммуникации и конфигурации) [1]

STATUS (W#16#....)	Описание
80D1	Получатель запросил управление потоком для прерывания текущей передачи, но не активировал передачу снова в течение времени ожидания. Эта ошибка также генерируется при аппаратном управлении потоком, если получатель в течение времени ожидания не распознает CTS.
80D2	Запрос на передачу был прерван, так как сигнал DSR не получен от DCE.
80E0	Сообщение завершено, так как принимающий буфер заполнен.
80E1	Сообщение завершено из-за ошибки четности.
80E2	Сообщение завершено из-за ошибки телеграммы.
80E3	Сообщение завершено из-за ошибки переполнения.
80E4	Сообщение отменено, так как указанная длина превышает суммарный размер буфера.
8180	Недопустимое значение для ID порта или ошибка инструкции Modbus_Comm_Load
8186	Недействительный адрес станции Modbus
8187	Недействительный указатель на блок данных MB_HOLD_REG: Область слишком маленькая.
818C	Недействительный указатель MB_HOLD_REG Область данных должна быть одной из следующих: • Классический DB • Массив элементарных типов данных в символьном или энергонезависимом DB • M память

Таблица 13-96 Коды условий для выполнения Modbus_Slave (ошибка протокола Modbus) [1]

STATUS (W#16#....)	Код ответа ведомого устройства	Ошибка протокола Modbus
8380	Нет ответа	Ошибка контр. суммы
8381	01	Код функции не поддерживается или не поддерживается в широковещательной рассылке
8382	03	Ошибка в длине данных

STATUS (W#16#....)	Код ответа ведомого устройства	Ошибка протокола Modbus
8383	02	Ошибка в адресе данных или адрес вне действительного диапазона DATA_PTR
8384	03	Ошибка в значении данных
8385	03	Значение диагностического кода данных не поддерживается (код функции 08)
8389		Недействительное определение для области данных: • Недопустимое значение для типа данных • Номер DB недействителен или отсутствует: – Недействительное значение DB – Номер DB отсутствует – Номер DB уже использует в другой области данных – DB с оптимизированным доступом – DB находится не в рабочей памяти • Недействительное значение длины • Наложение диапазонов адресов MODBUS, относящихся к одному типу данных MODBUS

[1] Наряду с перечисленными выше ошибками инструкций Modbus_Slave, базовые PtP коммуникационные инструкции также могут возвращать ошибки.

Примечание

Установка макс. длины записи для коммуникации PROFIBUS

Если коммуникационный модуль CM1243-5 выступает в роли ведущего устройства PROFIBUS для управления PROFIBUS устройством ET200SP или ET200MP, которое в свою очередь используется модуль точка-точка RS232, RS422 или RS485, переменная (тег) блока данных "max_record_len" должна быть явно установлена на значение 240 следующим образом:

Установить после выполнения Modbus_Comm_Load параметр max_record_len в разделе Send_P2P DB экземпляра (напр., "Modbus_Slave_DB".Send_P2P.max_record_len) на 240.

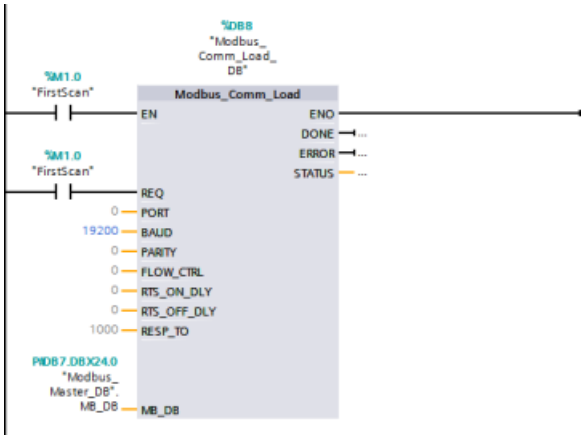
Явное присвоение значения для max_record_len необходимо только для PROFIBUS соединений. Для PROFINET соединений для max_record_len уже используется правильное значение.

13.5.3.5 Примеры Modbus RTU

Пример: Пример программы для ведущего устройства Modbus RTU

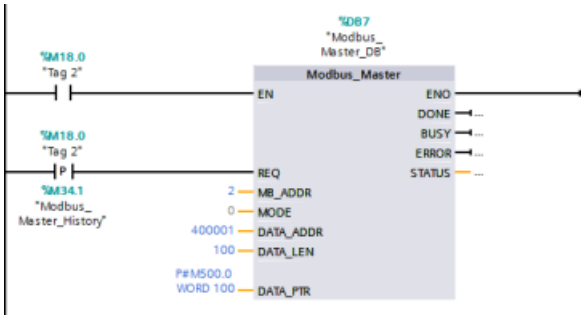
Modbus_Comm_Load инициализируется во время запуска при помощи флага первого цикла. Обработка Modbus_Comm_Load подобным образом должна выполняться только тогда, когда конфигурация последовательного порта не изменяется во время работы.

Сегмент 1: Конфигурировать/инициализировать коммуникационный порт RS485 модуля только один раз во время первого цикла.

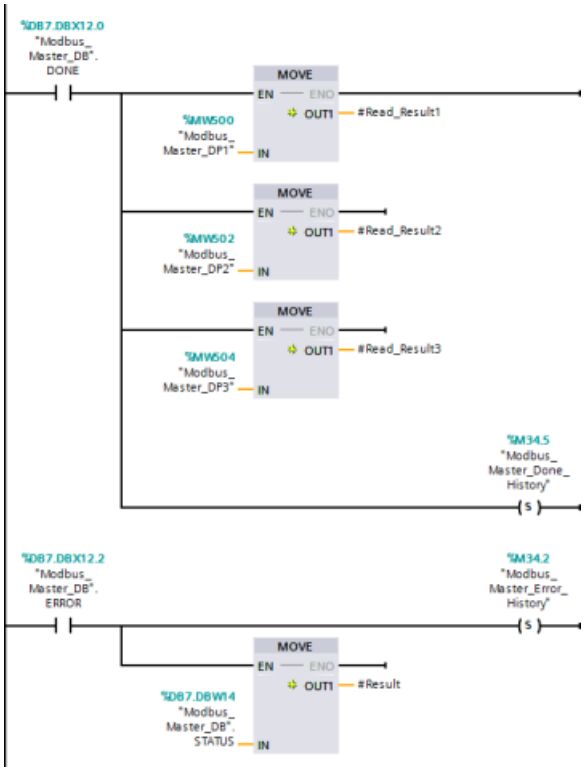


Одна инструкция Modbus_Master используется в OB программного цикла для обмена данными с ведомым устройством. Другие инструкции Modbus_Master могут использоваться в OB программного цикла, чтобы выполнить обмен данными с другими ведомыми устройствами, или один FB Modbus_Master может быть повторно использован, чтобы выполнить коммуникацию с другими ведомыми устройствами.

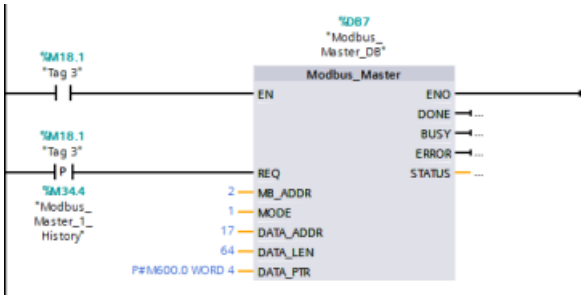
Сегмент 2: Выгрузить 100 слов данных регистра хранения по адресу 400001 на ведомом устройстве № 2 на адрес MW500-MW698.



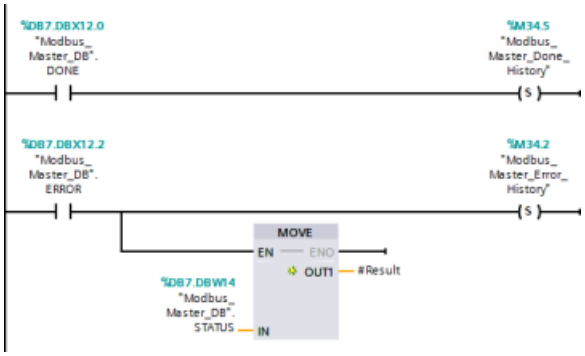
Сегмент 3: Переместить первые 3 слова данных регистра хранения, которые были выгружены на другой адрес, и установить бит истории DONE. Этот сегмент также устанавливает бит истории ERROR и сохраняет слово STATUS по другому адресу в случае ошибки.



Сегмент 4: Записать 64 бита данных из MW600-MW607 по адресам выходных битов с 00017 по 00081 на ведомом устройстве № 2.



Сегмент 5: Установить бит истории DONE после завершения операции записи. При возникновении ошибки программа устанавливает бит истории ERROR и сохраняет код STATUS.

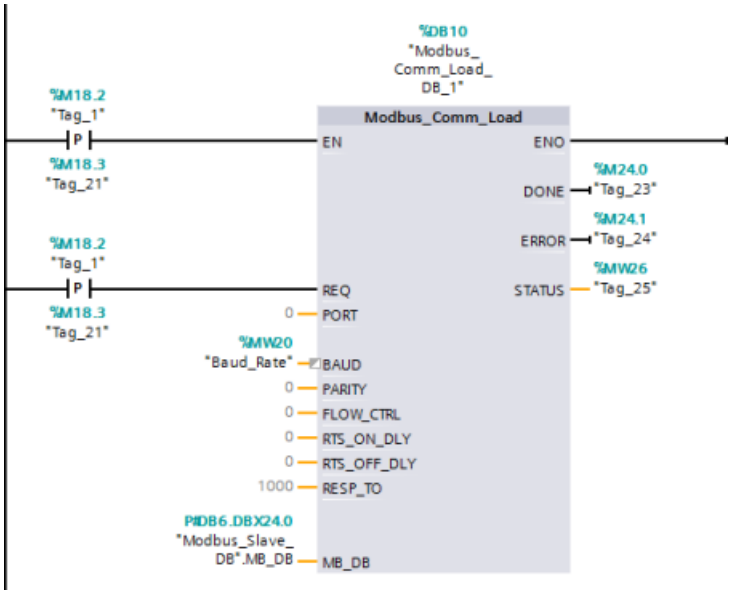


Пример: Пример программы для ведомого устройства Modbus

Инструкция MB_COMM_LOAD, показанная ниже, инициализируется каждый раз при активации "Tag_1".

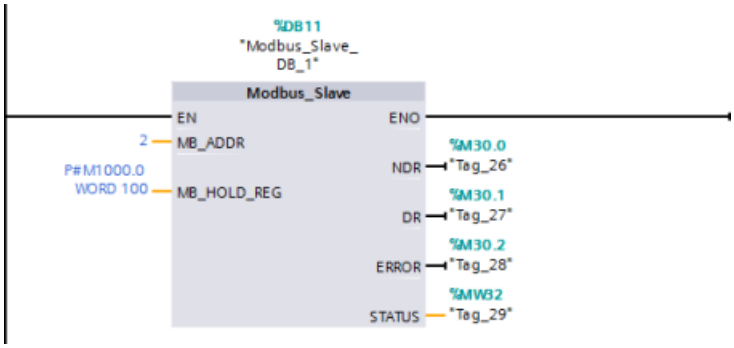
Обработка MB_COMM_LOAD подобным образом должна выполняться только тогда, когда конфигурация последовательного порта изменяется во время работы в результате конфигурирования устройства HMI.

Сегмент 1: Инициализировать параметры модуля RS485 каждый раз, когда они изменяются устройством HMI.



Инструкция MB_SLAVE, показанная ниже, находится в циклическом OB, который выполняется каждые 10 мс. Хотя это и не обеспечивает абсолютно быстрый ответ от ведомого устройства, но гарантирует хорошую производительность при 9600 бодах для коротких сообщений (до 20 байтов в запросе)

Сегмент 2: В каждом цикле проверять наличие запросов от ведущего устройства Modbus. Для регистра хранения Modbus сконфигурировано 100 слов с началом в MW1000.



13.6 PtP коммуникация в старых системах (только CM/CB 1241)

До STEP 7 версии V13 SP1 и CPU S7-1200 версии V4.1, коммуникационные инструкции "точка-точка" имели другие имена, и в некоторых случаях, несколько иные интерфейсы. Общие принципы для коммуникаций "точка-точка" (Страница 1119), а также конфигурирования порта (Страница 1123) и параметров (Страница 1139) применимы к обоим наборам инструкций. Обратиться к отдельным описаниям старых инструкций "точка-точка" для получения сведений о программировании.

Таблица 13-97 Общие классы ошибок

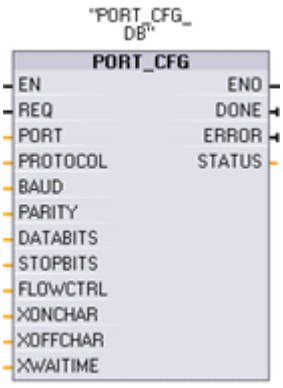
Описание класса	Классы ошибок	Описание
Конфигурация порта	80Ax	Для описания частых ошибок в конфигурации интерфейса
Конфигурация передачи	80Bx	Для описания частых ошибок в конфигурации передачи
Конфигурация приема	80Cx	Для описания частых ошибок в конфигурации приема
Время выполнения передачи	80Dx	Для описания частых ошибок в процессе передачи
Время выполнения приема	80Ex	Для описания частых ошибок в процессе приема
Обработка сигнала	80Fx	Для описания частых ошибок, связанных с обработкой сигналов

13.6.1 Старые инструкции точка-точка

13.6.1.1 PORT_CFG (динамическое конфигурирование коммуникационных параметров)

Таблица 13-98 Инструкция PORT_CFG (конфигурирование порта)

LAD/FBD	SCL	Описание

	`"PORT_CFG_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_uint_in_,`
	`PROTOCOL:=_uint_in_,`
	`BAUD:=_uint_in_,`
	`PARITY:=_uint_in_,`
	`DATABITS:=_uint_in_,`
	`STOPBITS:=_uint_in_,`
	`FLOWCTRL:=_uint_in_,`
	`XONCHAR:=_char_in_,`
	`XOFFCHAR:=_char_in_,`
	`WAITTIME:=_uint_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		Инструкция PORT_CFG позволяет изменять параметры порта, такие как скорость передачи, из программы.
		Можно настроить начальную статическую конфигурацию порта в свойствах конфигурации устройства или просто использовать значения по умолчанию. После с помощью инструкции PORT_CFG можно изменять конфигурацию в программе.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Изменения конфигурации с помощью PORT_CFG не сохраняются в CPU на постоянной основе. Параметры, сохраненные в конфигурации устройства, восстанавливаются при переходе CPU из RUN в STOP или при повторном включении. Дополнительную информацию можно найти здесь: Конфигурирование коммуникационных портов (Страница 1123) и Организация управления потоком (Страница 1125).

Таблица 13-99 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Активирует изменение конфигурации при положительном фронте данного входа. (значение по умолчанию: False)
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC. (значение по умолчанию: 0)
PROTOCOL	IN	UInt	0 - Протокол коммуникации "точка-точка" (значение по умолчанию) 1..n - Будущее определение для конкретных протоколов
BAUD	IN	UInt	Скорость передачи для порта (значение по умолчанию: 6): 1 = 300 бод, 2 = 600 бод, 3 = 1200 бод, 4 = 2400 бод, 5 = 4800 бод, 6 = 9600 бод, 7 = 19200 бод, 8 = 38400 бод, 9 = 57600 бод, 10 = 76800 бод, 11 = 115200 бод

Параметр и тип данных		Тип данных	Описание
PARITY	IN	UInt	Четность порта (значение по умолчанию: 1): 1 = без контроля четности, 2 = совпадение при контроле по чётности, 3 = совпадение при контроле по нечётности, 4 = контроль по единичному биту, 5 = контроль по нулевому биту
DATABITS	IN	UInt	Битов на символ (значение по умолчанию):1): 1 = 8 битов данных, 2 = 7 битов данных
STOPBITS	IN	UInt	Стоповые биты (значение по умолчанию: 1): 1 = 1 стоповый бит, 2 = 2 стоповых бита
FLOWCTRL	IN	UInt	Управление потоком (значение по умолчанию: 1): 1 = без управления потоком, 2 = XON/XOFF, 3 = аппаратный RTS всегда ВКЛ, 4 = аппаратный RTS с переключением
XONCHAR	IN	Char	Задает символ, который будет использоваться в качестве символа XON. Обычно это символ DC1 (16#11). Данный параметр анализируется только при активированном управлении потоком. (значение по умолчанию: 16#11)
XOFFCHAR	IN	Char	Задает символ, который будет использоваться в качестве символа XOFF. Обычно это символ DC3 (116#3). Данный параметр анализируется только при активированном управлении потоком. (значение по умолчанию: 16#13)
XWAITIME	IN	UInt	Определяет время ожидания символа XON после получения символа XOFF, или время ожидания сигнала CTS после активации RTS (от 0 до 65535 мс). Данный параметр анализируется только при активированном управлении потоком. (значение по умолчанию: 2000)
DONE	OUT	Bool	Для целого цикла выполнения TRUE, после того как последний запрос был выполнен без ошибок.
ERROR	OUT	Bool	Для целого цикла выполнения TRUE, после того как последний запрос был выполнен с ошибками.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

Таблица 13-100 Коды условий

STATUS (W#16#....)	Описание
80A0	Указанный протокол не существует.
80A1	Указанная скорость передачи не существует.
80A2	Указанная опция четности не существует.
80A3	Указанное число битов данных не существует.
80A4	Указанное число стоповых битов не существует.
80A5	Указанный вариант управления потоком не существует.
80A6	Время ожидания 0 и управление потоком активировано
80A7	XON и XOFF - это недопустимые значения (например, одинаковое значение)

13.6.1.2 SEND_CFG (конфигурация передачи)

Таблица 13-101 Инструкция SEND_CFG (конфигурация передачи)

LAD/FBD	SCL	Описание

	`"SEND_CFG_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_uint_in_,`
	`RTSONDLY:=_uint_in_,`
	`RTSOFFDLY:=_uint_in_,`
	`BREAK:=_uint_in_,`
	`IDLELINE:=_uint_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		Инструкция SEND_CFG обеспечивает динамическое конфигурирование параметров последовательной передачи для PtP коммуникационного порта. Любые сообщения в очереди в CM или CB сбрасываются, когда выполняется SEND_CFG.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Можно настроить начальную статическую конфигурацию порта в свойствах конфигурации устройства или просто использовать значения по умолчанию. После с помощью инструкции SEND_CFG можно изменять конфигурацию в программе.

Изменения конфигурации с помощью SEND_CFG не сохраняются в CPU на постоянной основе. Параметры, сохраненные в конфигурации устройства, восстанавливаются при переходе CPU из RUN в STOP или при повторном включении. См. Конфигурирование параметров передачи

Таблица 13-102 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Активирует изменение в конфигурации при положительном фронте на данном входе. (значение по умолчанию: False)
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC. (значение по умолчанию: 0)
RTSONDLY	IN	UInt	Количество миллисекунд, которые должны пройти после активации RTS, прежде чем будет выполнена передача данных Tx. Данный параметр действителен только при активированном аппаратном управлении потоком. Действительный диапазон находится между 0 и 65535 мс. Значение 0 выключает функцию. (значение по умолчанию: 0)
RTSOFFDLY	IN	UInt	Количество миллисекунд, которые должны пройти после передачи данных Tx, прежде чем будет выключен RTS: Данный параметр действителен только при активированном аппаратном управлении потоком. Действительный диапазон находится между 0 и 65535 мс. Значение 0 выключает функцию. (значение по умолчанию: 0)
BREAK	IN	UInt	Данный параметр указывает, что при запуске каждого сообщения на указанное количество битовых интервалов передается пауза. Максимум - это 65535 битовых интервалов до 8 секунд. Значение 0 выключает функцию. (значение по умолчанию: 12)

Параметр и тип данных		Тип данных	Описание
IDLELINE	IN	UInt	Данный параметр указывает, что перед запуском каждого сообщения на указанное количество битовых интервалов линия остается свободной. Максимум - это 65535 битовых интервалов до 8 секунд. Значение 0 выключает функцию. (значение по умолчанию: 12)
DONE	OUT	Bool	Для целого цикла выполнения TRUE, после того как последний запрос был выполнен без ошибок.
ERROR	OUT	Bool	Для целого цикла выполнения TRUE, после того как последний запрос был выполнен с ошибками.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

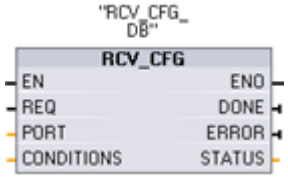
Таблица 13-103 Коды условий

STATUS (W#16#....)	Описание
80B0	Конфигурация прерывания передачи недопустима.
80B1	Время паузы превышает максимально допустимое значение.
80B2	Время простоя превышает максимально допустимое значение.

13.6.1.3 RCV_CFG (конфигурация приема)

Таблица 13-104 Инструкция RCV_CFG (конфигурация приема)

LAD/FBD	SCL	Описание

	`"RCV_CFG_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_uint_in_,`
	`CONDITIONS:=_struct_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		Инструкция RCV_CFG выполняет динамическое конфигурирование параметров последовательного приема для PtP коммуникационного порта. Данная инструкция выполняет конфигурацию условий, которые обозначают начало и конец полученного сообщения. Любые сообщения в очереди в CM или CB сбрасываются, когда выполняется RCV_CFG.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Можно настроить начальную статическую конфигурацию коммуникационного соединения в свойствах конфигурации устройства или просто использовать значения по умолчанию. После с помощью инструкции RCV_CFG можно изменять конфигурацию в программе.

Изменения конфигурации с помощью RCV_CFG не сохраняются в CPU на постоянной основе. Параметры, сохраненные в конфигурации устройства, восстанавливаются при

переходе CPU из RUN в STOP или при повторном включении. Дополнительную информацию см. в Конфигурирование параметров приема (Страница 1128).

Таблица 13-105 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Активирует изменение в конфигурации при положительном фронте на данном входе. (значение по умолчанию: False)
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC. (значение по умолчанию: 0)
CONDITIONS	IN	CONDITIONS	Структура данных CONDITIONS определяет условия начала и окончания сообщения, как описано ниже.
DONE	OUT	Bool	TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок.
ERROR	OUT	Bool	TRUE в течение одного цикла, после того как последний запрос был выполнен с ошибками.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

Условия начала для инструкции RCV_PTP

Инструкция RCV_PTP использует конфигурацию, заданную инструкцией RCV_CFG, чтобы определить начало и конец сообщений коммуникации "точка- точка". Начало сообщения определяется условиями начала. Начало сообщения определяется на основании одного или нескольких условий начала. Если указаны несколько условий начала, для запуска сообщения все условия должны быть соблюдены.

Описание условий для начала сообщения можно найти в "Конфигурирование параметров приема (Страница 1128)".

Структура типа данных параметра CONDITIONS, часть 1 (условия начала)

Таблица 13-106 Структура CONDITIONS для условий START

Параметр и тип данных		Тип данных	Описание
STARTCOND	IN	UInt	Указание условия начала (значение по умолчанию: 1) • 01H - Стартовый символ • 02H - Любой символ • 04H - Пауза на линии • 08H - Свободная линия • 10H - Последовательность символов 1 • 20H - Последовательность символов 2 • 40H - Последовательность символов 3 • 80H - Последовательность символов 4
IDLETIME	IN	UInt	Число битовых интервалов, требуемых для тайм-аута свободной линии. (значение по умолчанию: 40). Только в сочетании с условием "Свободная линия". От 0 до 65535
STARTCHAR	IN	Byte	Начальный символ для условия "Начальный символ". (значение по умолчанию: B#16#2)
SEQ[1].CTL	IN	Byte	Последовательность символов 1, игнорировать/сравнить управление для каждого символа: (значение по умолчанию: B#16#0) Это биты разрешения для каждого символа в последовательности начала • 01H - символ 1 • 02H - символ 2 • 04H - символ 3 • 08H - символ 4 • 10H - символ 5 Деактивация бита, связанного с символом, означает, что любой символ будет подходящим в этой позиции последовательности.
SEQ[1].STR	IN	Char[5]	Последовательность символов 1, стартовые символы (5 символов). Значение по умолчанию: 0
SEQ[2].CTL	IN	Byte	Последовательность символов 2, игнорировать/сравнить управление для каждого символа: (значение по умолчанию: B#16#0)
SEQ[2].STR	IN	Char[5]	Последовательность символов 2, стартовые символы (5 символов). Значение по умолчанию: 0
SEQ[3].CTL	IN	Byte	Последовательность символов 3, игнорировать/сравнить управление для каждого символа: Значение по умолчанию: B#16#0
SEQ[3].STR	IN	Char[5]	Последовательность символов 3, стартовые символы (5 символов). Значение по умолчанию: 0
SEQ[4].CTL	IN	Byte	Последовательность символов 4, игнорировать/сравнить управление для каждого символа: Значение по умолчанию: B#16#0
SEQ[4].STR	IN	Char[5]	Последовательность символов 4, стартовые символы (5 символов), значение по умолчанию: 0

Пример

Рассмотрим следующее принятое шестнадцатерично-кодированное сообщение: "**68** 10 aa **68** bb 10 aa 16". Сконфигурированные последовательности начала показаны в таблице ниже. Последовательности начала анализируются, когда первый 68H символ успешно получен. После получения четвертого символа (второй 68H) условие начала 1 считается выполненным. После выполнения условий начала выполняется анализ условий завершения.

Обработка последовательности начала может быть завершена в результате различных ошибок четности, синхронизации кадров или интервалов между символами. Эти ошибки приводят к тому, что сообщение не будет принято, так как условие начала не было выполнено.

Таблица 13-107 Условия начала

Условие начала	Первый символ	Первый символ +1	Первый символ +2	Первый символ +3	Первый символ +4
1	**68**H	xx	xx	**68**H	xx
2	10H	aaH	xx	xx	xx
3	dcH	aaH	xx	xx	xx
4	e5H	xx	xx	xx	xx

Условия завершения для инструкции RCV_PTP

Конец сообщения определяется указанными условиями завершения. Конец сообщения определяется первым появлением одного или нескольких сконфигурированных условий завершения. В разделе "Условия для начала сообщения" в пункте "Конфигурирование параметров приема (Страница 1128)" описываются условия окончания, которые могут быть сконфигурированы для инструкции RCV_CFG.

Выполнить конфигурацию условий завершения можно либо в свойствах коммуникационного интерфейса в конфигурации устройства, либо при помощи инструкции RCV_CFG. Каждый раз при переходе CPU из STOP в RUN параметры приема (условия начала и завершения) снова переключаются на настройки конфигурации устройства. При выполнении программой пользователя STEP 7 инструкции RCV_CFG, параметры устанавливаются на условия из RCV_CFG.

Структура типа данных параметра CONDITIONS, часть 2 (условия завершения)

Таблица 13-108 Структура CONDITIONS для условий END

Параметр	Тип параметра	Тип данных	Описание
ENDCOND	IN	UInt 0	Этот параметр задает условие для конца сообщения: • 01H - Время отклика • 02H - Время сообщения • 04H - Интервал между символами • 08H - Максимальная длина • 10H - N + LEN + M • 20H - Последовательность символов
MAXLEN	IN	UInt 1	Максимальная длина сообщения: Используется только в том случае, если выбрано условие завершения "Максимальная длина". от 1 до 1024 байт
N	IN	UInt 0	Байтовая позиция поля длины в сообщении. Используется только с условием завершения N + LEN + M. от 1 до 1022 байт
LENGTHSIZE	IN	UInt 0	Размер поля длины (1, 2 или 4 байта). Используется только с условием завершения N + LEN + M.
LENGTHM	IN	UInt 0	Указать количество символов после поля длины, которые не входят в значение поля длины. Используется только с условием завершения N + LEN + M. от 0 до 255 байт
RCVTIME	IN	UInt 200	Указать время ожидания для первого полученного символа. Инструкция приема заканчивается ошибкой, если в течение указанного времени не будет выполнен прием символа. Такое указание используется только при условии "Время отклика". (от 0 до 65 535 битовых интервалов, максимум 8 секунд) Этот параметр не является условием завершения, так как анализ прерывается, когда первый символ отклика получен. Это является условием завершения только в том смысле, что приводит к завершению работы инструкции приема при отсутствии какого-либо ответа, когда ответ ожидается. Необходимо определить отдельное условие завершения.
MSGTIME	IN	UInt 200	Указать время ожидания полного приема всего сообщения после получения первого символа. Данный параметр используется только в том случае, если выбрано условие "Тайм-аут сообщения". (от 0 до 65 535 мс)
CHARGAP	IN	UInt 12	Указать количество битовых интервалов между символами. Если количество битовых интервалов превысит указанное значение, условие завершения считается выполненным. Такое указание используется только при условии "Интервал между символами". (от 0 до 65 535 битовых интервалов, максимум 8 секунд)

Параметр	Тип параметра	Тип данных	Описание
SEQ.CTL	IN	Byte B#16#0	Последовательность символов 1, игнорировать/сравнить управление для каждого символа: Здесь указаны биты активации для каждого символа последовательность завершения. Символ 1 - бит 0, символ 2 - бит 1, ..., символ 5 - бит 4. Деактивация бита, связанного с символом, означает, что любой символ будет подходящим в этой позиции последовательности.
SEQ.STR	IN	Char[5] 0	Последовательность 1, начальный символ (5 символов)

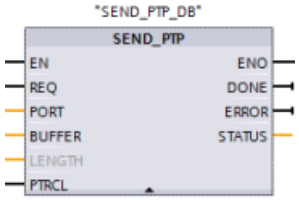
Таблица 13-109 Коды условий

STATUS (W#16#....)	Описание
80C0	Выбрано недопустимое условие начала
80C1	Недопустимое условие окончания или условие окончания не выбрано
80C2	Разрешено прерывание приема, и это невозможно.
80C3	Разрешено условие завершения по максимальной длине и максимальная длина равна 0 или > 1024.
80C4	Разрешена расчетная длина и N >= 1023
80C5	Разрешена расчетная длина и длина не равна 1, 2 или 4
80C6	Разрешена расчетная длина и значение M > 255.
80C7	Разрешена расчетная длина и она > 1024.
80C8	Разрешен тайм-аут отклика, и он равен нулю.
80C9	Разрешен тайм-аут межсимвольного интервала, и он равен нулю.
80CA	Разрешен тайм-аут свободной линии, и он равен нулю.
80CB	Последовательность завершения разрешена, но все символы, "игнорируются".
80CC	Последовательность начала (любая из 4) разрешена, но все символы, "игнорируются".

13.6.1.4 SEND_PTP (отправить данные из передающего буфера)

Таблица 13-110 Инструкция SEND_PTP (передача данных точка-точка)

LAD/FBD	SCL	Описание

	`"SEND_PTP_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_uint_in_,`
	`BUFFER:=_variant_in_,`
	`LENGTH:=_uint_in_,`
	`PTRCL:=_bool_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		Инструкция SEND_PTP запускает передачу данных и передает содержание назначенного буфера на коммуникационный интерфейс. Программа CPU продолжает работать, в то время как CM или CB отправляет данные с назначенной скоростью передачи. Одномоментно может быть активной только одна инструкция передачи. CM или CB возвращают ошибку, если вторая инструкция SEND_PTP обрабатывается, в то время как CM или CB уже передают сообщение.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Таблица 13-111 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Разрешение на запрошенную передачу при положительном фронте этого разрешающего входа. Таким образом, содержание буфера передается на соответствующий коммуникационный интерфейс "точка-точка". (значение по умолчанию: False)
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC. (значение по умолчанию: 0)
BUFFER	IN	Variant	Этот параметр указывает на начальный адрес передающего буфера. (значение по умолчанию: 0) **Примечание:** Логические данные или логические поля не поддерживаются.
LENGTH [1]	IN	UInt	Длина переданного кадра в байтах (значение по умолчанию: 0) При передаче сложной структуры всегда следует использовать длину 0.
PTRCL	IN	Bool	Зарезервировано для будущего использования
DONE	OUT	Bool	TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок.
ERROR	OUT	Bool	TRUE в течение одного цикла, после того как последний запрос был выполнен с ошибками.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

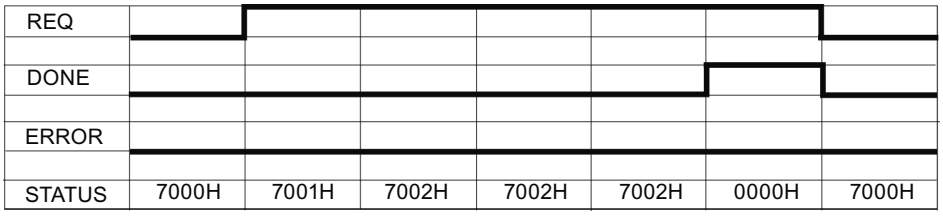
[1] Опциональный параметр: Кликнуть по стрелке у основания блока LAD/FBD, чтобы развернуть его и отобразить этот параметр.

При обработке инструкции передачи, выходы DONE и ERROR находятся в состоянии FALSE. После окончания инструкции передачи один из выходов DONE или ERROR

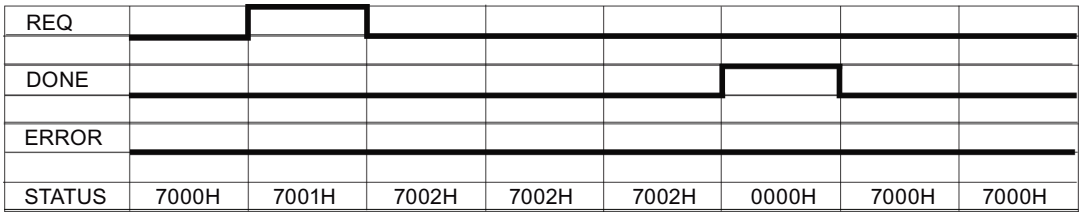
устанавливается на TRUE, чтобы сообщить о состояние инструкции передачи. Когда DONE или ERROR находятся в TRUE, выход STATUS допустим.

Инструкция возвращает состояние 16#7001, если коммуникационный интерфейс принимает переданные данные. Последующие выполнения SEND_PTP возвращают значение 16#7002, если CM или CB все еще заняты передачей. Когда передача завершена, CM или CB возвращают состояние процесса передачи как 16#0000 (при отсутствии ошибок). Последующее выполнение SEND_PTP с REQ = 0 возвращает состояние 16#7000 (не занято).

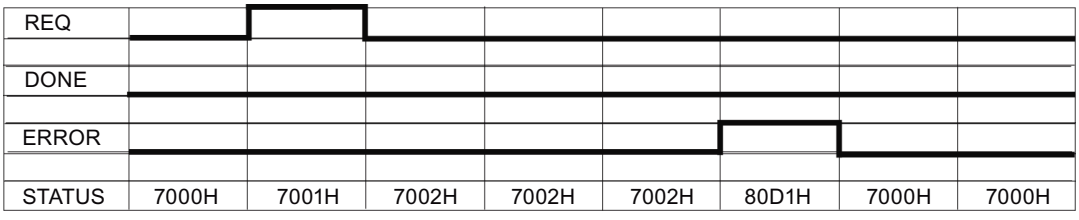
Следующая диаграмма показывает отношение выходных значений к REQ. Это предполагает, что инструкция вызывается периодически, чтобы проверить состояние процесса передачи. На диаграмме ниже предполагается, что инструкция вызывается в каждом цикле (представлено значениями STATUS).



Следующая диаграмма показывает, что параметры DONE и STATUS действительны только для одного цикла, если на линию REQ подается импульс (в течение одного цикла), чтобы инициировать процесс передачи.



Следующая диаграмма показывает отношение параметров DONE, ERROR и STATUS при ошибке.



Значения DONE, ERROR и STATUS допустимы, пока SEND_PTP выполняется снова с тем же экземплярным DB.

Таблица 13-112 Коды условий

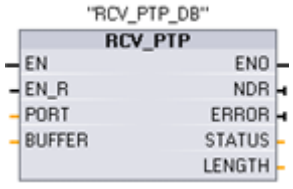
STATUS (W#16#....)	Описание
80D0	Новый запрос при активной передаче
80D1	Передача прервана из-за отсутствия CTS во время ожидания

STATUS (W#16#....)	Описание
80D2	Передача прервана из-за отсутствия DSR от DCE-устройства
80D3	Передача прервана из-за переполнения листа ожидания (передача более 1024 байтов)
80D5	Сигнал обратного смещения (обрыв провода)
833A	DB для параметра BUFFER не существует.

13.6.1.5 RCV_PTP (разрешить прием сообщений)

Таблица 13-113 Инструкция RCV_PTP (прием данных точка-точка)

LAD/FBD	SCL	Описание

	`"RCV_PTP_DB"(`
	`EN_R:=_bool_in_,`
	`PORT:=_uint_in_,`
	`BUFFER:=_variant_in_,`
	`NDR=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`LENGTH=>_uint_out_);`
		RCV_PTP поверяет сообщения, принятые в CM или CB. Если сообщение доступно, то оно передается CM или CB на CPU. Ошибка возвращает соответствующее значение STATUS.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Таблица 13-114 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
EN_R	IN	Bool	Если этот вход TRUE и доступно сообщение, то оно передается из CM или DB в BUFFER. Если EN_R = FALSE, то CM или CB проверяются на сообщения и выходы NDR, ERROR и STATUS обновляются, но сообщение в BUFFER не передается. (значение по умолчанию: 0)
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC. (значение по умолчанию: 0)
BUFFER	IN	Variant	Этот параметр указывает на начальный адрес принимающего буфера. Данный буфер должен иметь достаточно большой размер, чтобы принять телеграмму с максимальной длиной. Логические данные или логические поля не поддерживаются. (значение по умолчанию: 0)
NDR	OUT	Bool	Для целого цикла TRUE, если новые данные готовы и инструкция завершена без ошибок.
ERROR	OUT	Bool	Для целого цикла TRUE, после того как инструкция завершена с ошибкой.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)
LENGTH	OUT	UInt	Длина возвращенного сообщения в байтах (значение по умолчанию: 0)

Необходимо учитывать следующую связь между входом EN_R и буфером сообщений инструкции RCV_PTP:

Вход EN_R управляет копией полеченного сообщения в BUFFER.

Если вход EN_R =TRUE и доступно сообщение, то CPU передает сообщение от CM или CB в BUFFER и обновляет выходы NDR, ERROR, STATUS, и LENGTH.

Если EN_R = FALSE, то CPU проверяет CM или CB на сообщения и обновляет выходы NDR, ERROR, и STATUS, но не передает сообщение в BUFFER. (Следует помнить, что значение по умолчанию EN_R = FALSE.)

Рекомендуется установить EN_R на TRUE и использовать для управления выполнением инструкции RCV_PTP вход EN.

Значение STATUS является действительным, если NDR или ERROR в состоянии TRUE. Значение STATUS описывает причину завершения операции приема в CM или CB. Обычно это положительное значение, указывающее на то, что операция приема была успешна и что процесс приема завершен обычным образом. Если значение STATUS отрицательное (устанавливается старший бит шестнадцатеричного значения), операция приема была завершена из-за ошибки, например ошибки четности, синхронизации кадров или переполнения.

Каждый PtP коммуникационный интерфейс может поместить в буфер максимум 1024 байта. Это может быть одним большим сообщением или несколькими маленькими сообщениями. Если больше чем одно сообщение доступно в CM или CB, инструкция RCV_PTP возвращает самое старое доступное сообщение. Последующее выполнение инструкции RCV_PTP возвращает следующее самое старое доступное сообщение.

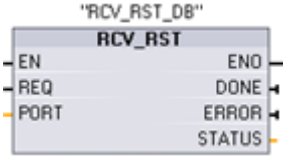
Таблица 13-115 Коды условий

STATUS (W#16#...)	Описание
0000	Буфер не существует
0094	Сообщение было завершено, так как была принята максимальная длина строки
0095	Сообщение было завершено из-за тайм-аута сообщения
0096	Сообщение было завершено из-за межсимвольного тайм-аута
0097	Сообщение было завершено из-за тайм-аута ответа
0098	Сообщение было завершено, так как было выполнено условие длины "N+LEN+M".
0099	Сообщение было завершено, так как было выполнено условие длины "N+LEN+M".
80E0	Сообщение завершено, так как принимающий буфер заполнен
80E1	Сообщение было завершено из-за ошибки четности
80E2	Сообщение было завершено из-за ошибки кадровой синхронизации
80E3	Сообщение было завершено из-за ошибки переполнения
80E4	Сообщение было завершено, так как расчетная длина превышает размер буфера.
80E5	Сигнал обратного смещения (обрыв провода)
833A	DB для параметра BUFFER не существует.

13.6.1.6 RCV_RST (удалить данные в принимающем буфере)

Таблица 13-116 Инструкция RCV_RST (сброс принимающего устройства)

LAD/FBD	SCL	Описание

	`"RCV_RST_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_uint_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		Инструкция RCV_RST очищает принимающий буфер в CM или CB.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Таблица 13-117 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Активирует удаление данных в принимающем буфере при положительном фронте разрешающего входа (значение по умолчанию: False)
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC. (значение по умолчанию: 0)
DONE	OUT	Bool	Для целого цикла TRUE обозначает, что последний запрос был выполнен без ошибок.
ERROR	OUT	Bool	TRUE обозначает, что последний запрос был выполнен с ошибками. Если этот выход TRUE, выход STATUS содержит соответствующие коды ошибок.
STATUS	OUT	Word	Код ошибки (значение по умолчанию: 0) Коды для состояния коммуникации см. в Общие параметры для операций точка-точка (Страница 1139).

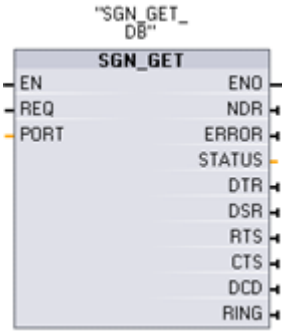
Примечание

С помощью инструкции RCV_RST можно убедиться, что буферы сообщений очищены после ошибки коммуникации или после изменения какого-либо параметра коммуникации, напр., скорости передачи. Выполнение RCV_RST очищает все внутренние буферы сообщений модуля. После очистки буферов сообщений можно быть уверенным, что программа при выполнении следующей инструкцию приема возвратит только новые сообщения, а не старые сообщения, относящиеся к времени до вызова RCV_RST.

13.6.1.7 SGN_GET (запросить сигналы RS232)

Таблица 13-118 Инструкция SGN_GET (получить сигналы RS232)

LAD/FBD	SCL	Описание

	`"SGN_GET_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_uint_in_,`
	`NDR=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`DTR=>_bool_out_,`
	`DSR=>_bool_out_,`
	`RTS=>_bool_out_,`
	`CTS=>_bool_out_,`
	`DCD=>_bool_out_,`
	`RING=>_bool_out_);`
		SGN_GET читает текущие состояния RS232 коммуникационных сигналов.
		Данная функция действительна только для RS232-CM.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Таблица 13-119 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Опрос состояния сигналов RS232 выполняется при положительном фронте на данном входе (значение по умолчанию: False)
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC.
NDR	OUT	Bool	Для целого цикла TRUE, если новые данные готовы и инструкция завершена без ошибок
ERROR	OUT	Bool	TRUE для целого цикла, после того как инструкция завершена с ошибкой.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)
DTR	OUT	Bool	Терминал данных готов, модуль готов (выход) Значение по умолчанию: False
DSR	OUT	Bool	Набор данных готов, участник обмена данными готов (вход). Значение по умолчанию: False
RTS	OUT	Bool	Запрос передачи, модуль готов к передаче (выход) Значение по умолчанию: False
CTS	OUT	Bool	Готов к передаче, участник обмена данными может принимать данные (вход). Значение по умолчанию: False
DCD	OUT	Bool	Носитель данных распознан, уровень сигнала получен (всегда False, не поддерживается)
RING	OUT	Bool	Индикатор вызова, сообщение входящего вызова (всегда False, не поддерживается)

Таблица 13-120 Коды условий

STATUS (W#16#....)	Описание
80F0	CM или CB - это RS-485 и нет доступных сигналов

13.6.1.8 SGN_SET (установить сигналы RS-232)

Таблица 13-121 Инструкция SGN_SET (установить сигналы RS-232)

LAD/FBD	SCL	Описание

	`"SGN_SET_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_uint_in_,`
	`SIGNAL:=_byte_in_,`
	`RTS:=_bool_in_,`
	`DTR:=_bool_in_,`
	`DSR:=_bool_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		Инструкция SGN_SET устанавливает состояния коммуникационных сигналов RS232.
		Данная функция действительна только для RS232-CM.

[1] При добавлении инструкции, STEP 7 автоматически создает DB.

Таблица 13-122 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Инструкция для установки сигналов RS232 запускается при положительном фронте этого входа (значение по умолчанию: False)
PORT	IN	PORT	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается во вкладке "Системные константы" таблицы переменных PLC. (значение по умолчанию: 0)
SIGNAL	IN	Byte	Выбирает сигналы для установки: (возможно несколько). Значение по умолчанию: 0 • 01H = RTS • 02H = DTR • 04H = DSR
RTS	IN	Bool	Запрос на передачу, модуль готов отправить значение для установки (true или false), значение по умолчанию: FALSE
DTR	IN	Bool	Терминал данных готов, модуль готов отправить значение для установки (true или false). Значение по умолчанию: FALSE
DSR	IN	Bool	Набор данных готов (действительно только для типа интерфейса DCE), не используется.
DONE	OUT	Bool	Для целого цикла выполнения TRUE, после того как последний запрос был выполнен без ошибок.

Параметр и тип данных		Тип данных	Описание
ERROR	OUT	Bool	Для целого цикла выполнения TRUE, после того как последний запрос был выполнен с ошибками.
STATUS	OUT	Word	Условие выполнения (значение по умолчанию: 0)

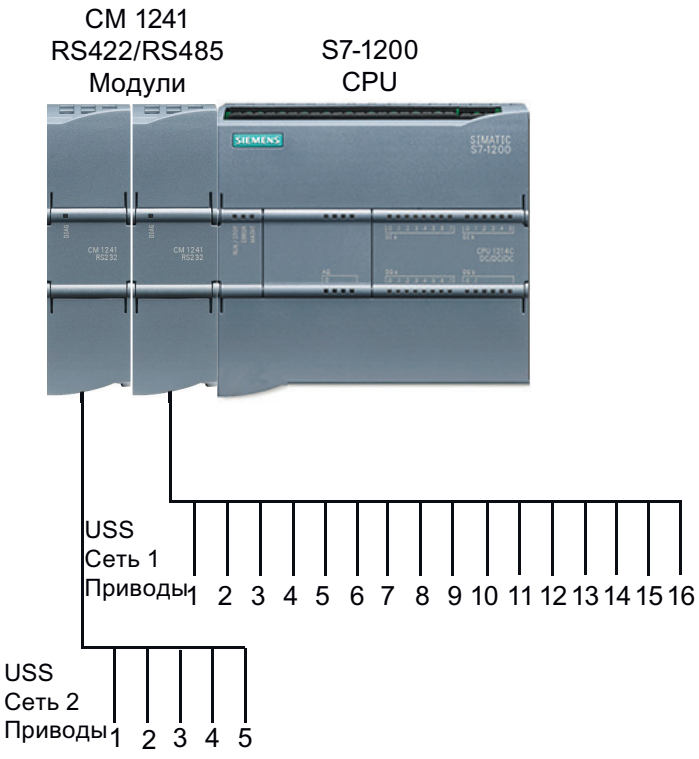
Таблица 13-123 Коды условий

STATUS (W#16#....)	Описание
80F0	CM или CB - это RS485 и никакие сигналы не могут быть установлены
80F1	Сигналы не могут быть установлены из-за аппаратного управления потоком
80F2	Нельзя установить DSR, так как модуль - DTE-устройство
80F3	Нельзя установить DTR, так как модуль - DTE-устройство

13.7 USS коммуникация в старых системах (только CM/CB 1241)

Инструкции USS управляют работой электроприводов, которые поддерживают протокол универсального последовательного интерфейса (USS). Можно использовать USS-инструкции, чтобы связаться с несколькими приводами через RS485-соединения с коммуникационными модулями CM 1241 RS485 или коммуникационной платой CB 1241 RS485. До трех модулей CM 1241 RS422/RS485 и одна плата CB 1241 RS485 могут быть установлены в S7-1200 CPU. Каждый порт RS485 может работать макс. с 16 приводами.

Протокол USS использует сеть ведущее устройство - ведомое устройство для коммуникации по последовательной шине. Ведущее устройство использует один параметр адреса, чтобы отправлять сообщение выбранному ведомому устройству. Ведомое устройство никогда не может выполнять отправку, если оно предварительно не получило соответствующий запрос. Прямой обмен сообщениями между отдельными ведомыми устройствами невозможен. Коммуникация USS работает в полудуплексном режиме. На изображении ниже показан сетевой график для примера приложения с приводом.



До STEP 7 версии V13 SP1 и CPU S7-1200 версии V4.1, USS коммуникационные инструкции имели другие имена, и в некоторых случаях, несколько иные интерфейсы. Общие концепции применимы к обоим типам инструкций. См. описания отдельные USS инструкций для получения информации о программировании.

13.7.1 Выбор версии USS-инструкций

Существует две версии инструкций USS в STEP 7:

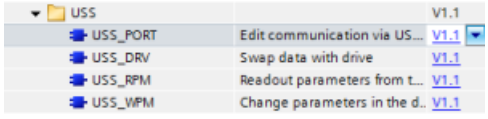
- Версия 2.0 изначально была доступна в STEP 7 Basic/Professional V13.
- Версия 2.1 доступна в STEP 7 Basic/Professional V13 SP1 и более поздних версий.

Учитывая совместимость и для упрощения миграции можно выбирать, какая версия инструкций будет вставлена в программу пользователя.

Нельзя использовать обе версии инструкций для одного модуля, но два различных модуля могут использовать различные версии инструкций.



Кликнуть по символу в окне задач с деревом инструкций, чтобы активировать заголовки и столбцы в дереве инструкций.



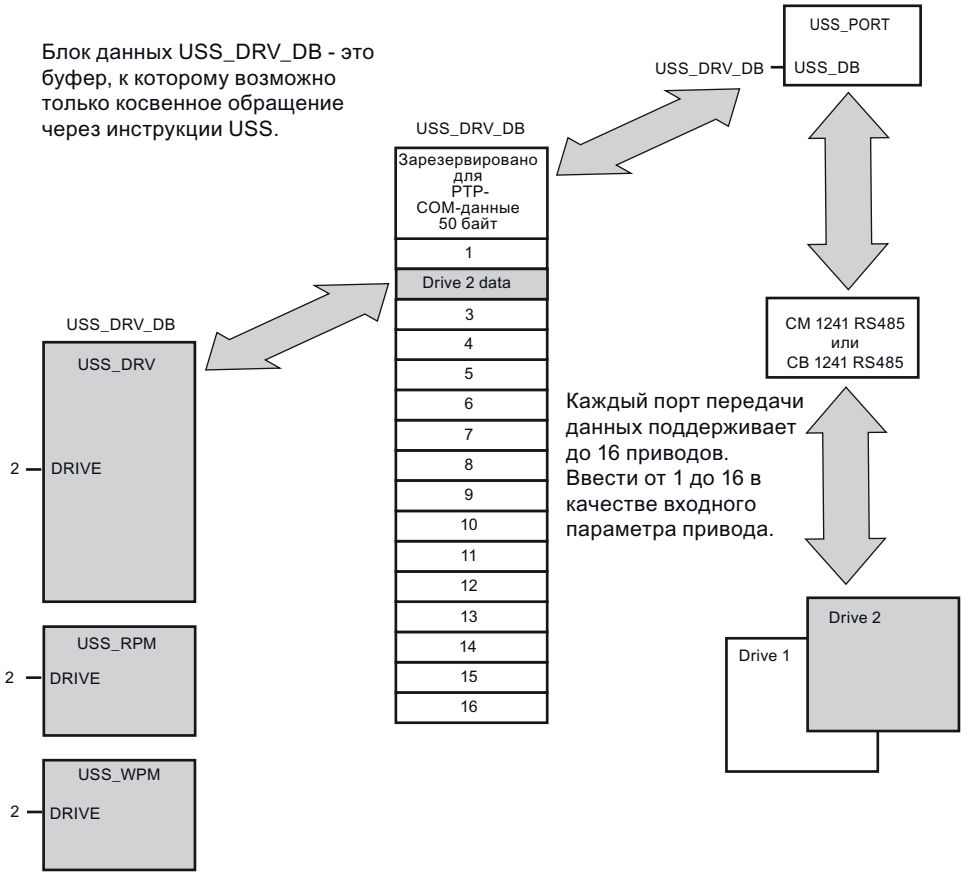
Для изменения версии USS инструкции, выбрать соответствующую версию в раскрывающемся списке. Можно выбрать группу или отдельные инструкции.

Если инструкция USS размещается через дерево инструкций в программе, то, в зависимости от выбранной инструкции USS, в дереве проекта создается новый экземпляр FB или FC. Для просмотра нового экземпляра FB или FC открыть в дереве проекта PLC_x > Программные блоки > Системные блоки > Программные ресурсы.

Чтобы узнать версию инструкции USS в программе, следует вызвать свойства в дереве проекта, а не свойства блочного элемента на экране в редакторе текстов программ. Выбрать в дереве проекта экземпляр FB или FC инструкции USS, кликнуть правой кнопкой мыши, выбрать "Свойства" и после страницу "Информация", чтобы отобразить номер версии инструкции USS.

13.7.2 Необходимые условия для использования протокола USS

Четыре инструкции USS используют 1 FB и 3 FC для поддержки протокола USS. Для каждой сети USS используется один блок данных экземпляра (DB) USS_PORT. Экземплярный блок данных USS_PORT содержит временное хранилище и буферы для всех приводов в USS-сети. Инструкции USS имеют общий доступ к информации в этом блоке данных.



Все приводы (макс. 16), подключенные к одному порту RS485, являются частью одной сети USS. Все приводы, подключенные к другому порту RS485, являются частью другой сети USS. Каждая сеть USS управляется при помощи уникального блока данных. Все инструкции, относящиеся одной сети USS, должны совместно использовать этот блок

данных. Он включает все инструкции "USS_DRV", "USS_PORT", "USS_RPM" и "USS_WPM" для управления всеми приводами в одной сети USS.

Инструкция USS_DRV является функциональным блоком (FB). Если добавить инструкцию USS_DRV в редактор текстов программ, в диалоговом окне "Опции вызова" придется для этого FB присвоить DB. Если речь идет о первой инструкции USS_DRV в этой программе для этой сети USS, можно принять стандартную инструкцию блока данных (или при необходимости изменить имя), и новый блок данных будет готов. Если же это не первая инструкция USS_DRV для этого канала, в диалоговом окне "Опции вызова" в выпадающем списке необходимо выбрать блок данных, который уже был назначен сети USS.

Во всех инструкциях USS_PORT, USS_RPM и USS_WPM речь идет о функциях (FC). Если вставлять эти функции в редакторе, DB не присваивается. Вместо этого входу USS_DB этих инструкций необходимо назначить соответствующий DB. Выполнить двойной клик по полю параметра и затем кликнуть по символу, чтобы отобразить доступные блоки данных.

Функция USS_PORT управляет коммуникацией между CPU и приводами посредством коммуникационного порта "точка-точка" (PtP) RS485. При каждом вызове этой функции обрабатывается коммуникация с одним приводом. В программе эта инструкция должна вызываться достаточно быстро для того, чтобы приводы не сообщали о превышении времени. Эта функция может быть вызвана из OB цикла основной программы или из любого OB прерываний.

Как правило, функция USS_PORT вызывается из OB циклического прерывания. Время цикла для OB циклического прерывания должно быть установлено приблизительно на половину минимального интервала вызова (например, обмен данными на скорости 1200 бод должен использовать время цикла в 350 мс или меньше).

Функциональный блок USS_DRV предоставляет вашей программе доступ к указанному приводу в сети USS. Его входы и выходы соответствуют состояниям и функциям управления привода. Если в сети имеется 16 приводов, в программе необходимо вызвать USS_DRV не менее 16 раз, т. е. один раз для каждого привода. Скорость вызова этих блоков зависит от необходимой скорости для управления функциями приводов.

Функциональный блок USS_DRV можно вызывать только из OB цикла главной программы.

⚠ ОСТОРОЖНО

Вызов инструкций USS из OB

Вызывать USS_DRV, USS_RPM и USS_WPM только из циклического OB главной программы. Функцию USS_PORT можно вызывать из любого OB, обычно она вызывается из OB циклического прерывания.

Не использовать инструкции USS_DRV, USS_RPM и USS_WPM в OB с более высоким приоритетом, чем соответствующая инструкция USS_PORT. Не следует, к примеру, вставлять USS_PORT в главную программу, а USS_RPM в OB циклического прерывания. Отказ предотвратить прерывание обработки USS_PORT может привести к неожиданным ошибкам, которые в свою очередь могут стать причиной телесных повреждений.

С помощью функций USS_RPM и USS_WPM считываются и записываются рабочие параметры удаленного привода. Эти параметры управляют внутренним принципом действия привода. Определение этих параметров см. в руководстве по эксплуатации привода. В программе может быть любое количество этих функций, но для одного

привода всегда должен быть активен только один запрос считывания или записи. Функции USS_RPM и USS_WPM можно вызывать только из циклического OB главной программы.

Расчет времени для коммуникации с приводом

Коммуникация с приводом выполняется асинхронно по отношению к циклу S71200. S7-1200 обычно проходит несколько циклов, прежде чем будет завершена коммуникация с приводом.

Промежуток USS_PORT является временем, необходимым для одной транзакции привода. В следующей таблице показаны минимальные интервалы для USS_PORT для каждой скорости передачи данных. Если вызывать функцию USS_PORT чаще, чем предписывает интервал USS_PORT, количество транзакций не увеличивается. Интервал тайм-аута привода - это промежуток времени, который доступен для транзакции, если для ее завершения из-за ошибок коммуникации требуется 3 попытки. По умолчанию библиотека USS протокола автоматически делает до 2 повторений на каждой транзакции.

Таблица 13-124 Расчет необходимого времени

Скорость передачи	Рассчитанный минимальный интервал для вызова USS_PORT (мс)	Тайм-аут интервала для сообщения на каждый привод (мс)
1200	790	2370
2400	405	1215
4800	212.5	638
9600	116.3	349
19200	68.2	205
38400	44.1	133
57600	36.1	109
115200	28.1	85

13.7.3 Старые инструкции USS

13.7.3.1 USS_PORT (обработка коммуникации через сеть USS)

Таблица 13-125 Инструкция USS_PORT

LAD/FBD	SCL	Описание

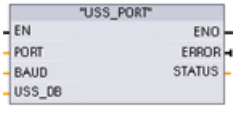
	`USS_PORT(`
	`  PORT:=_uint_in_,`
	`  BAUD:=_dint_in_,`
	`  ERROR=>_bool_out_,`
	`  STATUS=>_word_out_,`
	`  USS_DB:=_fbtref_inout_);`
		Инструкция USS_PORT обрабатывает коммуникацию по сети USS.

Таблица 13-126 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
PORT	IN	Port	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается на вкладке "Системные константы" таблицы переменных PLC.
BAUD	IN	DInt	Скорость передачи данных для коммуникации USS.
USS_DB	INOUT	USS_BASE	Имя блок данных экземпляра, который создается и инициализируется при вставке инструкции USS_DRV в программу.
ERROR	OUT	Bool	Если TRUE, этот выход указывает на ошибку и действительным является выход STATUS.
STATUS	OUT	Word	Значение состояния для запроса отображает результат цикла или инициализации. Дополнительная информация для отдельных кодов состояния доступна в переменной "USS_Extended_Error".

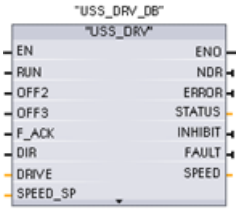
Обычно в программе для каждого коммуникационного порта PtP существует по одной инструкции USS_PORT, и каждый вызов этой инструкции управляет передачей данных на один привод этой сети или из него. Все функции USS, назначенные сети USS и коммуникационному порту PtP, должны использовать одинаковый экземплярный блок данных.

В программе инструкция USS_PORT должна выполняться достаточно часто, чтобы исключить тайм-аут привода. Обычно инструкция USS_PORT вызывается из OB циклического прерывания, чтобы предотвратить тайм-аут привода и сделать доступными последние обновления данных USS для вызовов USS_DRV.

13.7.3.2 Инструкция USS_DRV (обмен данными с приводом)

Таблица 13-127 Инструкция USS_DRV

LAD/FBD	SCL	Описание

Вид по умолчанию

 	
Расширенный вид

 	
	`"USS_DRV_DB"(`
	`RUN:=_bool_in_,`
	`OFF2:=_bool_in_,`
	`OFF3:=_bool_in_,`
	`F_ACK:=_bool_in_,`
	`DIR:=_bool_in_,`
	`DRIVE:=_usint_in_,`
	`PZD_LEN:=_usint_in_,`
	`SPEED_SP:=_real_in_,`
	`CTRL3:=_word_in_,`
	`CTRL4:=_word_in_,`
	`CTRL5:=_word_in_,`
	`CTRL6:=_word_in_,`
	`CTRL7:=_word_in_,`
	`CTRL8:=_word_in_,`
	`NDR=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`RUN_EN=>_bool_out_,`
	`D_DIR=>_bool_out_,`
	`INHIBIT=>_bool_out_,`
	`FAULT=>_bool_out_,`
	`SPEED=>_real_out_,`
	`STATUS1=>_word_out_,`
	`STATUS3=>_word_out_,`
	`STATUS4=>_word_out_,`
	`STATUS5=>_word_out_,`
	`STATUS6=>_word_out_,`
	`STATUS7=>_word_out_,`
	`STATUS8=>_word_out_);`
		Инструкция USS_DRV обменивается данными с приводом, создавая сообщения запроса и анализируя ответные сообщения привода. Для каждого привода необходимо использовать свой функциональный блок, но все функции USS, назначенные сети USS и коммуникационному порту PtP, должны использовать один экземплярный блок данных. При добавлении первой инструкции USS_DRV необходимо ввести имя блока данных. Затем указать ссылку на этот блок данных, который был создан при вставке первой инструкции.
		При вставке инструкции STEP 7 автоматически создает DB.

[1] LAD и FBD: Развернуть блок, чтобы просмотреть все параметры. Для этого кликнуть мышью по нижней области блока. Соединители параметров, отображенные серым цветом, являются опциональными, как и их параметрирование.

Таблица 13-128 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
RUN	IN	Bool	Стартовый бит привода: Если этот параметр TRUE, этот вход позволяет приводу работать с предустановленной скоростью. Если во время работы привода RUN меняется на false, двигатель выполняет выбег до остановки. Такое поведение отличается от отключения питания (OFF2) и от торможения двигателя (OFF3).
OFF2	IN	Bool	Бит "Выбег до останова": Если этот параметр FALSE, этот бит допускает инерционный выбег привода без торможения.
OFF3	IN	Bool	Быстрый стоповый бит: Если этот параметр FALSE, этот бит вызывает быстрый останов с помощью торможения привода.

Параметр и тип данных		Тип данных	Описание
F_ACK	IN	Bool	Бит квитирования ошибки: С помощью этого бита сбрасывается бит ошибки привода. Бит устанавливается после удаления ошибки, и таким образом привод распознает, что о предыдущей ошибке больше сообщать не надо.
DIR	IN	Bool	Управление направлением привода: Этот бит устанавливается, если привод должен вращаться вперед (при положит. SPEED_SP).
DRIVE	IN	USInt	Адрес привода: Этот вход является адресом привода USS. Действительный диапазон находится между приводом 1 и приводом 16.
PZD_LEN	IN	USInt	Длина в словах: Указывает на количество слов в данных PZD. Действительными значениями являются 2, 4, 6 или 8 слов. Стандартное значение равно 2.
SPEED_SP	IN	Real	Заданное значение скорости: Это скорость привода как процент заданной частоты. Положительное значение говорит о том, что привод вращается вперед (при истинном DIR). Действительный диапазон составляет от 200,00 до -200,00.
CTRL3	IN	Word	Управляющее слово 3: Значение, которое записывается в конфигурируемый пользователем параметр привода. Конфигурация выполняется в приводе (опциональный параметр).
CTRL4	IN	Word	Управляющее слово 4: Значение, которое записывается в конфигурируемый пользователем параметр привода. Конфигурация выполняется в приводе (опциональный параметр).
CTRL5	IN	Word	Управляющее слово 5: Значение, которое записывается в конфигурируемый пользователем параметр привода. Конфигурация выполняется в приводе (опциональный параметр).
CTRL6	IN	Word	Управляющее слово 6: Значение, которое записывается в конфигурируемый пользователем параметр привода. Конфигурация выполняется в приводе (опциональный параметр).
CTRL7	IN	Word	Управляющее слово 7: Значение, которое записывается в конфигурируемый пользователем параметр привода. Конфигурация выполняется в приводе (опциональный параметр).
CTRL8	IN	Word	Управляющее слово 8: Значение, которое записывается в конфигурируемый пользователем параметр привода. Конфигурация выполняется в приводе (опциональный параметр).
NDR	OUT	Bool	Новые данные готовы: При истинном значении этого параметра бит сообщает, что на выходе готовы данные нового запроса коммуникации.
ERROR	OUT	Bool	Возникла ошибка: Если TRUE, это указывает на ошибку, и действительным является выход STATUS. Все остальные выходы при возникновении ошибки устанавливаются на ноль. Об ошибках коммуникации сообщается только на выходах ERROR и STATUS инструкции USS_PORT.
STATUS	OUT	Word	Значение состояния запроса отображает результат цикла. Это не слово состояния, возвращаемое приводом.
RUN_EN	OUT	Bool	Работа разрешена: Этот бит сообщает о работе привода.
D_DIR	OUT	Bool	Направление привода: Этот бит сообщает, вращается ли привод вперед.
INHIBIT	OUT	Bool	Привод заблокирован: Этот бит сообщает о состоянии бита блокировки для привода.

Параметр и тип данных		Тип данных	Описание
FAULT	OUT	Bool	Ошибка привода: Этот бит сообщает, что в приводе возникла ошибка. Необходимо устранить неисправность и установить бит F_ACK, чтобы удалить этот бит.
SPEED	OUT	Real	Фактическое значение скорости привода (масштабированное значение от слова состояния 2 привода): значение скорости привода, как процент от заданной скорости.
STATUS1	OUT	Word	Слово состояния 1 привода: Данное значение включает в себя фиксированные биты состояния привода.
STATUS3	OUT	Word	Слово состояния 3 привода: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS4	OUT	Word	Слово состояния 4 привода: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS5	OUT	Word	Слово состояния 5 привода: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS6	OUT	Word	Слово состояния 6 привода: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS7	OUT	Word	Слово состояния 7 привода: Это значение содержит конфигурируемое пользователем слово состояния привода.
STATUS8	OUT	Word	Слово состояния 8 привода: Это значение содержит конфигурируемое пользователем слово состояния привода.

При первом выполнении инструкции USS_DRV, инициализируется указанный по адресу USS (параметр DRIVE) привод в экземплярном блоке данных. После инициализации последующие инструкции USS_PORT могут начинать коммуникацию с приводом по этому номеру привода.

Если изменить номер привода, CPU сначала необходимо перевести в состояние STOP и затем обратно в RUN, чтобы инициализировать экземплярный блок данных. Входные параметры конфигурируются в передающем буфере USS, а выходы, если имеются, считываются из "предыдущего" действительного буфера ответа. Во время выполнения инструкции USS_DRV передача данных не выполняется. После выполнения USS_PORT приводы выполняют обмен данными. USS_DRV конфигурирует только отправляемые сообщения и анализирует данные, которые возможно были получены в предыдущем запросе.

Направлением вращения привода можно управлять с помощью входа DIR (Bool) или с помощью знака (плюс или минус) на входе SPEED_SP (Real). В следующей таблице дается пояснение того, как взаимодействуют входы, чтобы задавать направление вращения привода, при условии, что двигатель вращается вперед.

Таблица 13-129 Взаимодействие параметров SPEED_SP и DIR

SPEED_SP	DIR	Направление вращения привода
Значение > 0	0	Назад
Значение > 0	1	Вперед
Значение < 0	0	Вперед
Значение < 0	1	Назад

13.7.3.3 Инструкция USS_RPM (считывание параметров из привода)

Таблица 13-130 Инструкция USS_RPM

LAD/FBD	SCL	Описание

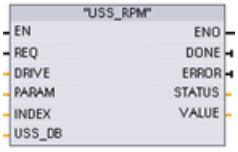
	`USS_RPM(REQ:=_bool_in_,`
	`DRIVE:=_usint_in_,`
	`PARAM:=_uint_in_,`
	`INDEX:=_uint_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`VALUE=>_variant_out_,`
	`USS_DB:=_fbtref_inout_);`
		Инструкция USS_RPM считывает параметр из привода. Все функции USS, назначенные сети USS и коммуникационному порту PtP, должны использовать один блок данных. USS_RPM должен вызываться из OB цикла главной программы.

Таблица 13-131 Типы данных для параметров

Тип параметра		Тип данных	Описание
REQ	IN	Bool	Запрос на передачу: Если REQ имеет значение TRUE, требуется новый запрос на чтение. Он игнорируется, если уже есть активный запрос для этого параметра.
DRIVE	IN	USInt	Адрес привода: DRIVE является адресом привода USS. Действительный диапазон находится между приводом 1 и приводом 16.
PARAM	IN	UInt	Номер параметра: PARAM указывает, какой параметр привода записывается. Диапазон для этого параметра составляет от 0 до 2047. В некоторых приводах старший значащий байт может получить доступ к значениям PARAM, больше, чем 2047. Дополнительная информация о доступе к расширенному диапазону находится в руководстве по эксплуатации привода.
INDEX	IN	UInt	Индекс параметра: INDEX указывает, в какой индекс параметра привода должна выполняться запись. Речь идет о 16-битном значении, в котором младший байт является фактическим значением индекса, с диапазоном (от 0 до 255). Старший байт также может использоваться приводом и зависит от типа привода. Дополнительная информация находится в руководстве по эксплуатации привода.
USS_DB	INOUT	USS_BASE	Имя блок данных экземпляра, который создается и инициализируется при вставке инструкции USS_DRV в программу.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDInt, Real	Это значение параметра, которое было считано, и оно действительно только в том случае, если бит DONE имеет истинное значение.
DONE[1]	OUT	Bool	Если этот параметр TRUE, на выходе VALUE будет запрошенное ранее значение параметра для чтения. Этот бит устанавливается, если инструкция USS_DRV распознает ответ считывания привода. Этот бит сбрасывается, если: Запрашиваются ответные данные, используя другой запрос , либо при втором из следующих двух вызовов USS_DRV.

Тип параметра		Тип данных	Описание
ERROR	OUT	Bool	Возникла ошибка: Если TRUE, ERROR указывает на ошибку и действительный выход STATUS. Все остальные выходы при возникновении ошибки устанавливаются на ноль. Об ошибках коммуникации сообщается только на выходах ERROR и STATUS инструкции USS_PORT.
STATUS	OUT	Word	STATUS показывает результат запроса на чтение. Дополнительная информация для отдельных кодов состояния доступна в переменной "USS_Extended_Error".

[1] Бит DONE указывает на то, что действительные данные были считаны из адресованного электропривода и отправлены CPU. Он не указывает на то, что библиотека USS в состоянии немедленно считать следующий параметр. На электропривод необходимо отправить пустой запрос PKW и квитировать его инструкцией, прежде чем канал параметров освободится для использования соответствующим приводом. Незамедлительный вызов функции USS_RPM или USS_WPM для конкретного электропривода приведет к ошибке "0x818A"

13.7.3.4 Инструкция USS_WPM (изменение параметров в приводе)

Примечание

EEPROM инструкции записи (для EEPROM в приводе USS)

Не использовать большое количество операций записи в EEPROM. Количество операций записи в EEPROM должно быть минимальным, чтобы продлить срок службы EEPROM.

Таблица 13-132 Инструкция USS_WPM

LAD/FBD	SCL	Описание

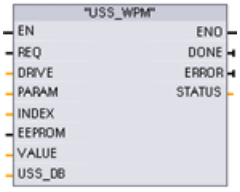
	`USS_WPM(REQ:=_bool_in_,`
	`DRIVE:=_usint_in_,`
	`PARAM:=_uint_in_,`
	`INDEX:=_uint_in_,`
	`EEPROM:=_bool_in_,`
	`VALUE:=_variant_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`USS_DB:=_fbtref_inout_);`
		Инструкция USS_WPM изменяет параметр в приводе. Все функции USS, присвоенные сети USS и коммуникационному порту PtP, должны использовать один блок данных.
		USS_WPM должна вызываться из OB цикла главной программы.

Таблица 13-133 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Запрос на передачу: Если REQ имеет значение TRUE, требуется новый запрос на запись. Он игнорируется, если уже есть активный запрос для этого параметра.
DRIVE	IN	USInt	Адрес привода: DRIVE является адресом привода USS. Действительный диапазон находится между приводом 1 и приводом 16.

Параметр и тип данных		Тип данных	Описание
PARAM	IN	UInt	Номер параметра: PARAM указывает, какой параметр привода записывается. Диапазон для этого параметра составляет от 0 до 2047. В некоторых приводах старший значащий байт может получить доступ к значениям PARAM, больше, чем 2047. Дополнительная информация о доступе к расширенному диапазону находится в руководстве по эксплуатации привода.
INDEX	IN	UInt	Индекс параметра: INDEX указывает, в какой индекс параметра привода должна выполняться запись. Речь идет о 16-битном значении, в котором младший байт является фактическим значением индекса, с диапазоном (от 0 до 255). Старший байт также может использоваться приводом и зависит от типа привода. Дополнительная информация находится в руководстве по эксплуатации привода.
EEPROM	IN	Bool	Сохранение в EEPROM привода: Если TRUE, транзакция параметра для записи сохраняется в привод в EEPROM привода. Если FALSE, записанное значение сохраняется временно и утрачивается при следующем включении привода.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDInt, Real	Значение параметра, в который будет выполняться запись. Оно должно быть действительным при смене состояния REQ.
USS_DB	INOUT	USS_BASE	Имя блок данных экземпляра, который создается и инициализируется при вставке инструкции USS_DRV в программу.
DONE[1]	OUT	Bool	Если TRUE, DONE указывает на то, что вход VALUE был записан в привод. Этот бит устанавливается, если инструкция USS_DRV распознает ответ записи привода. Этот бит сбрасывается, когда запрашиваются ответные данные, используя другой запрос USS_WPM, либо при втором из следующих двух вызовов USS_DRV.
ERROR	OUT	Bool	Если TRUE, ERROR указывает на ошибку и действительный выход STATUS. Все остальные выходы при возникновении ошибки устанавливаются на ноль. Об ошибках коммуникации сообщается только на выходах ERROR и STATUS инструкции USS_PORT.
STATUS	OUT	Word	STATUS показывает результат запроса записи. Дополнительная информация для отдельных кодов состояния доступна в переменной "USS_Extended_Error".

[1] Бит DONE указывает на то, что действительные данные были считаны из адресованного электропривода и отправлены CPU. Он не указывает на то, что библиотека USS в состоянии немедленно считать следующий параметр. На электропривод необходимо отправить пустой запрос PKW и квитировать его инструкцией, прежде чем канал параметров освободится для использования соответствующим приводом. Незамедлительный вызов функции USS_RPM или USS_WPM для конкретного электропривода приведет к ошибке "0x818A"

13.7.4 Старые коды состояния USS

Коды состояния USS-инструкций возвращаются на выходе STATUS для функций USS.

Таблица 13-134 STATUS-коды [1]

STATUS (W#16#....)	Описание
0000	Ошибки отсутствуют
8180	Длина ответа привода не соответствовала символам, принятым от привода. Номер привода, в котором возникла ошибка, возвращается в переменной "USS_Extended_Error". Описание расширенных ошибок дается под этой таблицей.
8181	Параметр VALUE не относится к типу данных Word, Real или DWord.
8182	Пользователь ввел значение параметра типа Word, а ответ от привода получен в формате DWord или Real.
8183	Пользователь ввел значение параметра типа DWord или Real, а ответ от привода получен в формате Word.
8184	В ответной телеграмме привода была неверная контрольная сумма. Номер привода, в котором возникла ошибка, возвращается в переменной "USS_Extended_Error". Описание расширенных ошибок дается под этой таблицей.
8185	Недопустимый адрес привода (действительный диапазон адресов для привода: от 1 до16)
8186	Заданное значение скорости вне действительного диапазона (действительный диапазон заданного значения для скорости: от -200 % до 200 %).
8187	Неверный номер привода ответил на отправленный запрос. Номер привода, в котором возникла ошибка, возвращается в переменной "USS_Extended_Error". Описание расширенных ошибок дается под этой таблицей.
8188	Недопустимая длина PZD в словах (действительный диапазон = 2, 4, 6 или 8 слов)
8189	Указана недопустимая скорость передачи данных
818A	Канал запроса для параметра используется другим запросом для этого привода.
818B	Привод не отреагировал на запросы и повторы. Номер привода, в котором возникла ошибка, возвращается в переменной "USS_Extended_Error". Описание расширенных ошибок дается под этой таблицей.
818C	Привод возвратил расширенную ошибку на запрос параметра. Описание расширенных ошибок дается под этой таблицей.
818D	Привод возвратил ошибку "Несанкционированный доступ" на запрос параметра. Дополнительную информацию о том, как можно ограничить доступ к параметру, см. в руководстве по эксплуатации своего привода.
818E	Привод не был инициализирован. Этот код ошибки выдается в USS_RPM или USS_WPM, если инструкция USS_DRV не была вызвана для этого привода хотя бы один раз. Таким образом предотвращается то, что инициализация в первом цикле USS_DRV перезапишет имеющийся запрос считывания или записи параметров, так как при этом привод инициализируется как новая запись. Чтобы устранить эту ошибку, вызвать инструкцию USS_DRV для этого привода.
80Ax-80Fx	Определенные ошибки возвращены из FB для PtP-коммуникаций, вызванных USS-библиотекой. - Эти коды ошибок не изменяются USS-библиотекой и определены в описаниях PtP-инструкции.

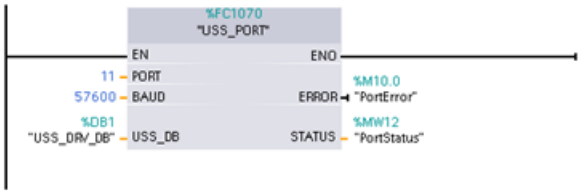
[1] Наряду с перечисленными выше ошибками USS инструкций, базовые PtP коммуникационные инструкции также могут возвращать ошибки.

Для различных кодов STATUS дополнительная информация предоставляется в переменной "USS_Extended_Error" экземплярного блока данных USS_DRV. Для шестн. кодов параметра STATUS 8180, 8184, 8187 и 818B, переменнаяUSS_Extended_Error

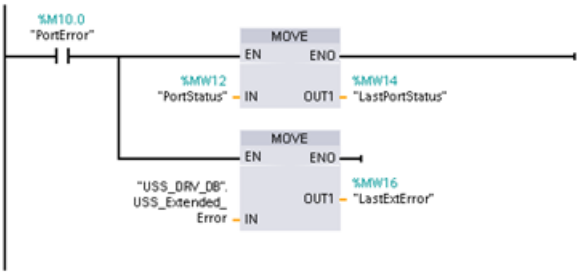
содержит номер привода, на котором возникла ошибка обмена данными. Для шестн. кодов STATUS параметра 818C, переменная USS_Extended_Error содержит код ошибки привода, возвращаемый приводом при использовании инструкции USS_RPM или USS_WPM.

Пример: Оповещение об ошибках коммуникации

Об ошибках коммуникации (STATUS = 16#818B) сообщает только инструкция USS_PORT, но не инструкция USS_DRV. Пример: Если сегмент должным образом не завершен, то привод может перейти в RUN, но инструкция USS_DRV покажет "0" для всех выходных параметров. В этом случае ошибку коммуникации можно обнаружить только с помощью инструкции USS_PORT. Так как эта ошибка отображается только один цикл, необходимо будет добавить соответствующую логику захвата. Процесс показа в примере ниже. В этом примере, если бит ошибки инструкции USS_PORT имеет значение TRUE, то значения STATUS и USS_Extended_Error помещаются в M область памяти. Номер привода помещается в переменную USS_Extended_Error, если значение кода STATUSравно шестнадцатеричным 8180, 8184, 8187 или 818B.



Сегмент 1 Состояние порта "PortStatus" и расширенные значения кодов ошибок "USS_DRV_DB".USS_Extended_Error действительны только в течение одного цикла программы. Значения должны быть зарегистрированы для дальнейшей обработки.



Сегмент 2 Контакт "PortError" запускает сохранение значения "PortStatus" в "LastPortStatus" и значения "USS_DRV_DB".USS_Extended_Error в "LastExtError".

Доступ для чтения и записи к внутренним параметрам привода

Приводы USS поддерживают доступ по чтению и записи к своим внутренним параметрам. Эта функция позволяет выполнять удаленное управление и конфигурирование привода. Операции доступа к параметрам привода могут перестать работать из-за ошибок, таких как выход значения из диапазона или недопустимых запросов в текущем режиме работы привода. Привод генерирует код ошибки, который возвращается в переменной "USS_Extended_Error". Этот код ошибки действует только для последнего выполнения инструкции USS_RPM или USS_WPM. Код ошибки привода сохраняется в переменной "USS_Extended_Error", если шестн. значение STATUS code равно 818C. Код ошибки "USS_Extended_Error" зависит от варианта привода. Описание расширенных кодов ошибок для функций чтения и записи для параметров см. в руководстве по эксплуатации привода.

13.7.5 Общие требования при настройке привода с использованием старой инструкции USS

При настройке привода с использованием старой инструкции USS действуют следующие общие требования:

- Для приводов должно быть настроено использование четырех слов PKW.
- Приводы могут конфигурироваться для 2, 4, 6 или 8 слов PZD.
- Количество слов PZD в приводе должно соответствовать входу PZD_LEN инструкции USS_DRV привода.
- Скорость передачи всех приводов должна соответствовать входу BAUD инструкции USS_PORT .
- Привод должен быть настроен для дистанционного управления.
- Привод должен быть настроен для уставки частоты по USS на COM канале.
- Адрес привода должен быть установлен в диапазоне 1 - 16. Этот адрес должен соответствовать входу DRIVE на блоке USS_DRV привода.
- Для управления направлением привода необходимо настроить использование полярности заданного значения привода.
- Оконечная нагрузка сети RS485 должна быть включена надлежащим образом.

Общее USS-подключение и настройка привода являются одинаковыми для USS инструкций (V4.1) и старых USS-инструкций (V4.0 и ранее). Дополнительную информацию см. в Пример: Общее USS-подключение и настройка привода (Страница 1192).

13.8 Modbus TCP коммуникация в старых системах

13.8.1 Обзор

До STEP 7 версии V13 SP1 и CPU S7-1200 версии V4.1, Modbus TCP коммуникационные инструкции имели другие имена, и в некоторых случаях, несколько иные интерфейсы. Общие концепции применимы к обоим типам инструкций. См. описания отдельных Modbus TCP инструкций для получения информации о программировании.

13.8.2 Выбор версии Modbus TCP инструкций

Следующие версии инструкций Modbus TCP доступны в STEP 7:

- Старая версия 2.1: совместима со всеми моделями CPU и CM
- Старая версия 3.1: совместима со всеми моделями CPU и CM
- Версия 4.2: совместима с CPU от версии V4.0 и CM от версии V2.1
- Версия 5.1: совместима с CPU от версии V4.2 и CM от версии V2.1

Учитывая совместимость и для упрощения миграции можно выбирать, какая версия инструкций будет вставлена в программу пользователя.

Вызвать на карте задач "Инструкция" инструкции Modbus TCP в группе Коммуникационный процессор в пункте "Другие".

Для изменения версии инструкции Modbus TCP, выбрать соответствующую версию в раскрывающемся списке. Можно выбрать группу или отдельные инструкции.



Если инструкция Modbus TCP размещается через дерево инструкций в программе, то в дереве проекта создается новый экземпляр FB. Для просмотра нового экземпляра FB открыть в дереве проекта PLC_x > Программные блоки > Системные блоки > Программные ресурсы.

Чтобы узнать версию инструкции Modbus TCP в программе, следует вызвать свойства в дереве проекта, а не свойства блочного элемента на экране в редакторе текстов программ. Выбрать в дереве проекта экземпляр FB инструкции Modbus TCP, кликнуть правой кнопкой мыши, выбрать "Свойства" и после страницу "Информация", чтобы отобразить номер версии инструкции Modbus TCP.

13.8.3 Старые инструкции Modbus TCP

13.8.3.1 MB_CLIENT (обмен данными с использованием PROFINET в качестве клиента Modbus TCP)

Таблица 13-135 Инструкция MB_CLIENT

LAD/FBD	SCL	Описание

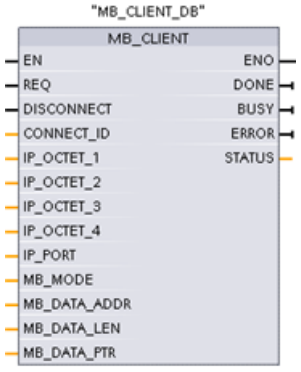
	`"MB_CLIENT_DB"(`
	`REQ:=_bool_in_,`
	`DISCONNECT:=_bool_in_,`
	`CONNECT_ID=_uint_in_,`
	`IP_OCTET_1:=_byte_in_,`
	`IP_OCTET_2:=_byte_in_,`
	`IP_OCTET_3:=_byte_in_,`
	`IP_OCTET_4:=_byte_in_,`
	`IP_PORT:=_uint_in_,`
	`MB_MODE:=_usint_in_,`
	`MB_DATA_ADDR:=_udint_in_,`
	`MB_DATA_LEN:=_uint_in_,`
	`DONE=>_bool_out_,`
	`BUSY=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`MB_DATA_PTR:=_variant_inout_);`
		MB_CLIENT выполняет обмен данными как Modbus TCP клиент через соединение PROFINET на S7-1200 CPU. Никакой дополнительный коммуникационный аппаратный модуль не требуется.
		MB_CLIENT может выполнить соединение клиент-сервер, отправить запрос функции Modbus, получить ответ и управлять разъединением от сервера Modbus TCP.

Таблица 13-136 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	In	Bool	FALSE = Нет коммуникационного запроса Modbus TRUE = Запрос на обмен данными с сервером Modbus TCP
DISCONNECT	IN	Bool	Параметр DISCONNECT позволяет программе управлять соединением с сервером Modbus. Если DISCONNECT = 0 и соединение не существует, то MB_CLIENT пытается выполнить соединение с назначенным IP-адресом и номером порта. Если DISCONNECT = 1 и соединение существует, то предпринимается попытка отключения. Пока этот вход активирован, никакая другая операция не будет предпринята.
CONNECT_ID	IN	UInt	Параметр CONNECT_ID должен однозначно идентифицировать каждое соединение внутри устройства PLC. Каждый уникальный экземпляр инструкции MB_CLIENT или MB_SERVER должен содержать уникальный параметр CONNECT_ID.
IP_OCTET_1	IN	USInt	IP-адрес сервера Modbus TCP: Октет 1 8-разрядная часть 32-разрядного IP-адреса (IPv4) Modbus TCP сервера, к которому клиент подключится и выполнит обмен по протоколу Modbus TCP.
IP_OCTET_2	IN	USInt	IP-адрес сервера Modbus TCP: Октет 2
IP_OCTET_3	IN	USInt	IP-адрес сервера Modbus TCP: Октет 3
IP_OCTET_4	IN	USInt	IP-адрес сервера Modbus TCP: Октет 4
IP_PORT	IN	UInt	Значение по умолчанию = 502: номер IP порта сервера, к которому клиент попытается подключиться и выполнить обмен, используя TCP/IP.
MB_MODE	IN	USInt	Выбор режима: Назначает тип запроса (чтение, запись или диагностика). Дополнительную информацию можно найти в таблице функций Modbus ниже.
MB_DATA_ADDR	IN	UDInt	Начальный адрес Modbus: Назначает начальный адрес данных, к которым должен получить доступ MB_CLIENT. Действительные адреса можно найти в таблице функций Modbus ниже.
MB_DATA_LEN	IN	UInt	Длина данных Modbus: Назначает количество бит или слов, к которым будет обращаться этот запрос. Действительные значения длины можно найти в таблице функций Modbus ниже.
MB_DATA_PTR	IN_OUT	Variant	Указатель на регистр данных Modbus: Регистр помещает в буфер данные, идущие в или приходящие от сервера Modbus. Указатель должен назначать не оптимизированный глобальный DB или адрес в M памяти.
DONE	OUT	Bool	Бит DONE = TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок.
BUSY	OUT	Bool	• 0 - нет обрабатываемых инструкций MB_CLIENT • 1 - обрабатывается инструкция MB_CLIENT
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение TRUE, после того, как последнее выполнение MB_CLIENT было завершен с ошибкой. Код ошибки в параметре STATUS действителен только в цикле, где ERROR = TRUE.
STATUS	OUT	Word	Условие выполнения

Параметр REQ

FALSE = Нет коммуникационного запроса Modbus
TRUE = Запрос на обмен данными с сервером Modbus TCP

Если никакой из экземпляров MB_CLIENT не активен и параметр DISCONNECT=0, то при REQ=1 запускается новый запрос Modbus. Если соединение еще не установлено, то выполняется новое соединение.

Если тот же экземпляр MB_CLIENT будет выполняться снова с DISCONNECT=0 и REQ=1 перед завершением текущего запроса, то последующая передача Modbus не будет выполнена. Однако, как только текущий запрос завершен, новый запрос может быть обработан, если MB_CLIENT выполняется с REQ=1.

Если текущий коммуникационный запрос MB_CLIENT завершен, бит DONE устанавливается в TRUE на один цикл. Бит DONE может использоваться в качестве временного окна для упорядочения множественных запросов MB_CLIENT.

Примечание

Согласованность входных данных во время обработки MB_CLIENT

Как только Modbus клиент инициирует Modbus операцию, все входные состояния сохраняются в системе и затем сравниваются при каждом последующем вызове. Сравнение используется, чтобы определить, был ли этот определенный вызов инициатором активного клиентского запроса. Можно выполнить несколько вызовов MB_CLIENT , используя общий экземплярный блок данных.

Важно, чтобы входы не были изменены в течение промежутка времени, когда операция MB_CLIENT активно обрабатывается. Если это правило не соблюдается, то MB_CLIENT не может определить активный экземпляр.

Параметры MB_MODE и MB_DATA_ADDR выбирают коммуникационные функции Modbus

Инструкция MB_DATA_ADDR назначает начальный адрес Modbus для обращения. Инструкция MB_CLIENT использует вместо входа кода функции вход MB_MODE.

Комбинация из значений MB_MODE и MB_DATA_ADDR определяет код функции, используемый в собственной телеграмме Modbus. В нижеследующей таблице показано отношение между параметром MB_MODE, функцией Modbus и диапазоном адресов Modbus.

Таблица 13-137 Функции Modbus

MB_MODE	Функция Modbus	Длина данных	Операция и данные	MB_DATA_ADDR
0	01	От 1 до 2000	Чтение выходных битов: От 1 до 2000 бит за запрос	От 1 до 9999
0	02	От 1 до 2000	Чтение входных битов: От 1 до 2000 бит за запрос	От 10001 до 19999
0	03	От 1 до 125	Чтение регистра хранения: От 1 до 125 слов за запрос	От 40001 до 49999 или от 400001 до 465535

MB_MODE	Функция Modbus	Длина данных	Операция и данные	MB_DATA_ADDR
0	04	От 1 до 125	Чтение входных слов: От 1 до 125 слов за запрос	От 30001 до 39999
1	05	1	Запись выходного бита: Один бит на запрос	От 1 до 9999
1	06	1	Запись регистра хранения: 1 слово на запрос	От 40001 до 49999 или от 400001 до 465535
1	15	От 2 до 1968	Запись нескольких выходных битов: От 2 до 1968 бит за запрос	От 1 до 9999
1	16	От 2 до 123	Запись нескольких регистров хранения: От 2 до 123 слов за запрос	От 40001 до 49999 или от 400001 до 465535
2	15	От 1 до 1968	Запись одного или нескольких выходных битов: От 1 до 1968 бит за запрос	От 1 до 9999
2	16	От 1 до 123	Запись одного или нескольких регистров хранения: От 1 до 123 слов за запрос	От 40001 до 49999 или от 400001 до 465535
11	11	0	Считывание слова состояния и счетчика событий коммуникации с сервером. Слово состояния указывает на "занято" (0 – не занято, 0xFFFF – занято). Счетчик событий инкрементируется при каждой удачной обработке сообщения. Как параметр MB_DATA_ADDR, так и параметр MB_DATA_LEN из MB_CLIENT игнорируются для этой функции.	
80	08	1	Проверить состояние сервера с помощью диагностического кода 0x0000 (петлевой тест, эхо запроса от сервера) 1 слово на запрос	
81	08	1	Сброс счетчика событий сервера посредством диагностического кода данных 0x000A 1 слово на запрос	
3 до 10, 12 до 79, 82 до 255			Зарезервировано	

Примечание

MB_DATA_PTR назначает буфер для хранения данных чтения/записи в Modbus TCP сервер

Буфер данных может быть расположен в не оптимизированном глобальном DB или M области памяти.

Для буфера в M памяти следует использовать стандартный формат указателя Any. Это формат P# "Адрес бита" "Тип данных" "Длина". Пример: P#M1000.0 WORD 500.

Параметр MB_DATA_PTR назначает коммуникационный буфер

- Функции коммуникации MB_CLIENT:
  - Читать и записывать 1-битовые данные из адресов сервера Modbus (от 00001 до 09999)
  - Читать и записывать 1-битовые данные из адресов сервера Modbus (от 10001 до 19999)
  - Читать 16-разрядные данные слов из адресов сервера Modbus (от 30001 до 39999) и (от 40001 до 49999)
  - Записывать 16-разрядные данные слов в адреса сервера Modbus (от 40001 до 49999)
- Данные размерностью слово или бит передаются в/из DB или M буфера памяти, назначенного MB_DATA_PTR.
- Если DB назначен, как буфер MB_DATA_PTR, то необходимо присвоить типы данных всем элементам данных DB.
  - 1-битовый тип данных Bool представляет один Modbus адрес бита
  - 16-битовые типы данных отдельного слова, такие как WORD, UInt и Int представляют один Modbus адрес слова
  - 32-разрядные типы данных двойного слова, такие как DWORD, DInt и Real представляют два Modbus адреса слова
- MB_DATA_PTR могут быть назначены сложные элементы DB, такие как
  - Стандартные массивы
  - Именованные структуры, где каждый элемент уникален.
  - Именованные сложные структуры, где каждый элемент имеет уникальное имя и 16- или 32-битный типа данных.
- Не требуется, чтобы области данных MB_DATA_PTR были в том же глобальном блоке данных (или M области памяти). Можно назначить один блок данных для чтения по Modbus, другой блок данных для записи по Modbus или один блок данных для каждой станции MB_CLIENT.

Подключение нескольких клиентов

Клиент Modbus TCP может поддерживать параллельные соединения вплоть до максимального количества соединений для открытых коммуникационных соединений пользователя (Open user communication), разрешенных для устройства PLC. Общее количество соединений для устройства PLC, включая клиентов и сервер Modbus TCP, не должно превышать максимальное количество поддерживаемых открытых коммуникационных соединений пользователя (Страница 807). Соединения Modbus TCP могут совместно использоваться подключениям типа клиента и/или сервера.

Отдельные клиентские соединения должны следовать данным правилам:

- Каждое соединение MB_CLIENT должно использовать свой DB экземпляра.
- Каждое соединение MB_CLIENT должно назначать уникальный IP-адрес сервера.

- Каждое соединение MB_CLIENT должно назначать уникальный ID соединения.
- Уникальные номера портов IP могут потребоваться в зависимости от конфигурации сервера.

ID соединения должен быть уникальным для каждого отдельного соединения. Это означает, что отдельный, уникальный ID соединения может использоваться только с каждым отдельным экземплярным DB. Таким образом, экземплярный DB и ID соединения составляют пару и должны быть уникальными для каждого соединения.

Таблица 13-138 Доступные для пользователя статические переменные блока данных экземпляра MB_CLIENT

Переменная	Тип данных	По умолчанию	Описание
Blocked_Proc_Timeout	Real	3.0	Время (в секундах), в течение которого ожидается заблокированный экземпляр клиента Modbus, прежде чем этот экземпляр будет удален со статусом ACTIVE. Это может произойти, например, когда был выставлен клиентский запрос и впоследствии приложение прекращает выполнять клиентскую функцию до полного завершения запроса. Максимальный предел для S7-1200 составляет 55 секунд.
MB_Unit_ID	Word	255	Идентификатор устройства Modbus: Сервер Modbus TCP адресуется с помощью своего IP-адреса. Поэтому параметр MB_UNIT_ID не используется при адресации Modbus TCP. Параметр MB_UNIT_ID соответствует полю адреса ведомого устройства в протоколе Modbus RTU. Если сервер Modbus TCP используется в качестве шлюза к протоколу Modbus RTU, MB_UNIT_ID может использоваться для идентификации ведомого устройства, подключенного к последовательной сети. Параметр MB_UNIT_ID мог бы использоваться для передачи запроса по корректному адресу ведомого устройства Modbus RTU. Некоторые устройства Modbus TCP могут потребовать, чтобы параметр MB_UNIT_ID для инициализации находился в ограниченном диапазоне значений.
RCV_TIMEOUT	Real	2,0	Время в секундах, в течение которого MB_CLIENT ожидает сервер, чтобы ответить на запрос.
Connected	Bool	0	Указывает на установление соединения с назначенным сервером: 1 = соединение установлено , 0 = соединение не установлено

Таблица 13-139 Ошибка протокола MB_CLIENT

STATUS (W#16#....)	Код ответа Modbus клиенту (B#16#)	Ошибка протокола Modbus
8381	01	Код функции не поддерживается
8382	03	Ошибка в длине данных
8383	02	Ошибка в адресе данных или доступ вне диапазона адресов MB_HOLD_REG
8384	03	Ошибка в значении данных
8385	03	Значение диагностического кода данных не поддерживается (код функции 08)

Таблица 13-140 Коды состояния выполнения MB_CLIENT [1]

STATUS (W#16#....)	Ошибка параметра MB_CLIENT
7001	MB_CLIENT ожидает ответа сервера Modbus на запрос подключения или отключения на назначенном порте TCP. Это сигнализируется только для первого выполнения операции подключения или отключения.
7002	MB_CLIENT ожидает ответа сервера Modbus на запрос подключения или отключения на назначенном порте TCP. Это будет сигнализировано для любого последующего выполнения при ожидании завершения операции подключения или отключения.
7003	Операция отключения успешно завершена (действует только для одного цикла PLC)
80C8	Сервер не ответил в назначенное время. MB_CLIENT должен принять ответ, используя ID транзакции, который был первоначально передан в течение назначенного времени, или будет возвращена данная ошибка. Проверить соединение с серверным устройством Modbus. Об этой ошибке сообщается только после того, как были предприняты сконфигурированные повторы (если применимо).
8188	Неправильное значение режима
8189	Недопустимое значение для адреса данных
818A	Недопустимое значение для длины данных
818B	Недействительный указатель на область DATA_PTR Здесь речь может идти о комбинации MB_DATA_ADDRESS + MB_DATA_LEN.
818C	Указатель на оптимизированною область DATA_PTR (при этом речь должна идти о не оптимизированной области DB или области M памяти)
8200	Порт занят обработкой существующего запроса Modbus.
8380	Принятый кадр Modbus имеет неправильный формат, или было получено слишком мало байтов.
8387	Назначенный параметр ID соединения отличается от ID, использованного для предыдущих запросов. Может быть только один ID соединения, используемый в каждом экземплярном DB для MB_CLIENT. Это также и внутренняя ошибка, если ID протокола Modbus TCP, полученный от сервера, не 0.
8388	Сервер Modbus возвратил количество данных, которое отличается от запрошенного количества. Это относится только к функциям Modbus 15 и 16.

[1] В дополнение к упомянутым выше ошибкам MB_CLIENT, ошибки могут быть возвращены из базовых коммуникационных инструкций T блока (TCON, TDISCON, TSEND и TRCV (Страница 864)).

13.8.3.2 MB_SERVER (обмен данными с использованием PROFINET в качестве сервера Modbus TCP)

Инструкция MB_SERVER выполняет обмен данными как сервер Modbus TCP через порт PROFINET на CPU S7-1200. Инструкция MB_SERVER обрабатывает запросы на установление соединения клиента Modbus TCP, получает и обрабатывает запросы Modbus и отправляет ответы.

Для использования инструкции дополнительный аппаратный модуль не требуется.

ВНИМАНИЕ
Указания по безопасности Каждый клиент в сети имеет доступ на чтение и запись к входам и выходам образа процесса, а также к области блоков данных и битовой памяти, которые определены в регистре хранения Modbus. Эта опция должна ограничить доступ к IP-адресу для предотвращения несанкционированных операций чтения и записи. Однако следует обратить внимание на то, что общий адрес может быть использован и для несанкционированного доступа.

Таблица 13-141 Инструкция MB_SERVER

LAD/FBD	SCL	Описание

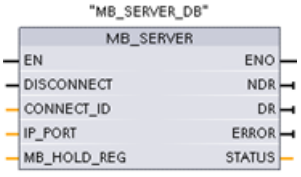
	`"MB_SERVER_DB"(`
	`DISCONNECT:=_bool_in_,`
	`CONNECT_ID:=_uint_in_,`
	`IP_PORT:=_uint_in_,`
	`NDR=>_bool_out_,`
	`DR=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`MB_HOLD_REG:=_variant_inout_);`
		MB_SERVER выполняет обмен данными как Modbus TCP сервер через соединение PROFINET на S7-1200 CPU. Никакой дополнительный коммуникационный аппаратный модуль не требуется.
		MB_SERVER может принять запрос на соединение с клиентом Modbus TCP, получить запрос функции Modbus и отправить ответное сообщение.

Таблица 13-142 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
DISCONNECT	IN	Bool	MB_SERVER пытается установить "пассивное" соединение с устройством партнера. Это означает, что сервер пассивно "прислушивается" к запросу на установление TCP соединения от любого запрашивающего IP-адреса. Если DISCONNECT = 0 и соединения отсутствуют, то может быть установлено пассивное соединение. Если DISCONNECT = 1 и соединение существует, то инициируется операция отключения. Это позволяет программе выбирать момент для подтверждения соединения. Пока этот вход активирован, никакая другая операция не будет предпринята.
CONNECT_ID	IN	UInt	CONNECT_ID однозначно идентифицирует каждое соединение в устройстве PLC. Каждый уникальный экземпляр инструкции MB_CLIENT или MB_SERVER должен содержать уникальный параметр CONNECT_ID.
IP_PORT	IN	UInt	Значение по умолчанию = 502: Номер IP порт для идентификации порта, который будет контролироваться на предмет запроса на установление соединения от клиента Modbus. Следующие номера TCP портов запрещены для пассивного соединения MB_SERVER: 20, 21, 25, 80, 102, 123, 5001, 34962, 34963 и 34964.

Параметр и тип данных		Тип данных	Описание
MB_HOLD_REG	IN_OUT	Variant	Указатель на Modbus регистр хранения MB_SERVER: Регистр хранения должен быть либо не оптимизированным глобальным DB, либо адресом в M памяти. Эта область памяти используется, чтобы хранить значения, к которым может обращаться Modbus клиент, используя регистровые функции Modbus: 3 (чтение), 6 (запись) и 16 (запись).
NDR	OUT	Bool	Новые данные готовы: 0 = нет новых данных, 1 = указывает на то, что новые данные были записаны клиентом Modbus
DR	OUT	Bool	Считывание данных: 0 = нет чтения данных, 1 = указывает на то, что данные были считаны клиентом Modbus
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение TRUE, после того, как выполнение MB_SERVER было завершен с ошибкой. Код ошибки в параметре STATUS действителен только в цикле, где ERROR = TRUE.
STATUS	OUT	Word	Условие выполнения

MB_SERVER позволяет входящим кодам функций Modbus (1, 2, 4, 5 и 15) читать/записывать биты/слова непосредственно в образах процесса ввода/вывода S7-1200 CPU. Для кодов функций передачи данных (3, 6 и 16), параметр MB_HOLD_REG должен быть определен как тип данных, длиннее, чем байт. Следующая таблица показывает согласование адресов Modbus для образа процесса в CPU.

Таблица 13-143 Согласование адресов Modbus для образа процесса

Функции Modbus						S7-1200	
Коды	Функция	Область данных	Диапазон адресов			Область данных	Адрес CPU
01	Читать биты	Вывод	1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7
02	Читать биты	Ввод	10001	до	18192	Образ процесса по вводу	От I0.0 до I1023.7
04	Чтение слов	Ввод	30001	до	30512	Образ процесса по вводу	От IW0 до IW1022
05	Записать бит	Вывод	1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7
15	Записать биты	Вывод	1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7

Коды функций входящих сообщений Modbus (3, 6, и 16) читают/записывают слова в регистре хранения Modbus, который может быть адресной область в M памяти или блоком данных. Тип регистра хранения определяется параметром MB_HOLD_REG.

Примечание

Назначение параметра MB_HOLD_REG

Регистр хранения Modbus может быть либо не оптимизированным глобальным DB, либо адресом в M памяти.

Для регистра хранения Modbus в M памяти следует использовать стандартный формат указателя Any. Это формат P# "Адрес бита" "Тип данных" "Длина". Пример: P#M1000.0 WORD 500.

Следующая таблица показывает примеры согласования адресов Modbus с регистрами хранения, используемые для кодов функций Modbus 03 (чтение слов), 06 (запись слов) и 16 (запись слов). Фактический верхний предел адресов DB определен максимальным объемом рабочей памяти и объемом M памяти для каждой из моделей CPU.

Таблица 13-144 Примеры для согласования адресов Modbus с адресами в памяти CPU

Адрес Modbus	**Примеры для параметра MB_HOLD_REG**		
	P#M100.0 Word 5	**P#DB10.DBx0.0 Word 5**	**"Recipe".ingredient**
40001	MW100	DB10.DBW0	"Recipe".ingredient[1]
40002	MW102	DB10.DBW2	"Recipe".ingredient[2]
40003	MW104	DB10.DBW4	"Recipe".ingredient[3]
40004	MW106	DB10.DBW6	"Recipe".ingredient[4]
40005	MW108	DB10.DBW8	"Recipe".ingredient[5]

Подключения нескольких серверов

Может быть создано несколько подключений серверов. Это позволяет одиночному PLC устанавливать параллельные соединения с несколькими клиентами Modbus TCP.

Сервер Modbus TCP может поддерживать параллельные соединения вплоть до максимального количества соединений для открытых коммуникационных соединений пользователя (Open user communication), разрешенных для устройства PLC. Общее количество соединений для устройства PLC, включая клиентов и сервер Modbus TCP, не должно превышать максимальное количество поддерживаемых открытых коммуникационных соединений пользователя (Страница 807). Соединения Modbus TCP могут совместно использоваться подключениям типа клиента и/или сервера.

Отдельные серверные соединения должны следовать данным правилам:

- Каждое соединение MB_SERVER должно использовать свой DB экземпляра.
- Каждое соединение MB_SERVER должно быть установлено с уникальным номером IP-порта. Для каждого порта поддерживается только одно соединение.

- Каждое соединение MB_SERVER должно использовать уникальный ID соединения.
- MB_SERVER должен вызываться индивидуально для каждого соединения (с соответствующим экземплярным DB).

ID соединения должен быть уникальным для каждого отдельного соединения. Это означает, что отдельный, уникальный ID соединения может использоваться только с каждым отдельным уникальным экземплярным DB. Таким образом, экземплярный DB и ID соединения составляют пару и должны быть уникальными для каждого соединения.

Таблица 13-145 Коды диагностических функций Modbus

Функции диагностики Modbus для MB_SERVER		
Коды	Подфункция	Описание
08	0x0000	Вывод данных запроса эхо-контроля: Инструкция MB_SERVER возвращает Modbus клиенту эхо полученного слова данных.
08	0x000A	Очистка счетчика событий коммуникации: Инструкция MB_SERVER очищает счетчик событий коммуникации, который используется для Modbus функции 11.
11		Вызов счетчика событий коммуникации: Инструкция MB_SERVER использует внутренний счетчик событий коммуникации для регистрации количества выполненных запросов чтения и записи Modbus, которые отправляются на сервер Modbus. Значение счетчика не увеличивается при выполнении функций 8 или функции 11. Значение счетчика не увеличивается и при запросах, которые приводят к ошибке связи. Функция широковещательной передачи недоступна для Modbus TCP, так как в любой момент имеется только одно соединение клиент-сервер.

Переменные MB_SERVER

В таблице ниже приведены общедоступные статические переменные в экземплярном блоке данных MB_SERVER, которые могут быть использованы в программе.

Таблица 13-146 Общедоступные статические переменные MB_SERVER

Переменная	**Тип данных**	**По умолчанию**	**Описание**
HR_Start_Offset	Word	0	Задает начальный адрес регистра хранения Modbus
Request_Count	Word	0	Количество всех полученных этим сервером запросов
Server_Message_Count	Word	0	Количество запросов, полученных для этого определенного сервера
Xmt_Rcv_Count	Word	0	Количество операций по передаче или приему данных, в которых произошли ошибки. Кроме того, увеличивается на единицу, если принято сообщение, которое является недопустимым сообщением Modbus.
Exception_Count	Word	0	Ошибки Modbus, требующие возвращенного исключения
Success_Count	Word	0	Количество запросов, полученных для этого определенного сервера без ошибок протокола.
Connected	Bool	0	Указывает, существует ли или нет соединение с назначенным клиентом. 1 = соединение установлено , 0 = соединение не установлено

Программа может записывать значения в HR_Start_Offset и управлять операциями сервера Modbus. Другие переменные могут считываться для контроля состояния Modbus.

HR_Start_Offset

Адреса регистра хранения Modbus начинаются с 40001. Эти адреса соответствуют начальному адресу регистра хранения в памяти целевой системы. Однако можно использовать переменную HR_Start_Offset, чтобы сконфигурировать начальный адрес для регистра хранения Modbus, отличный от 40001.

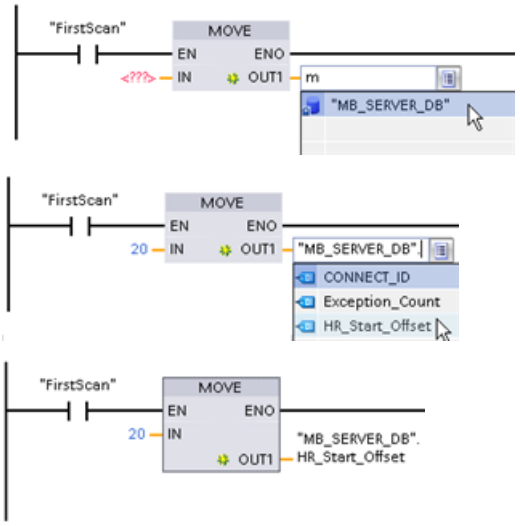
Можно, например, сконфигурировать регистр хранения с началом в MW100 и длиной в 100 слов. При помощи смещения 20 можно задать начальный адрес регистра хранения 40021 вместо 40001. Любой адрес меньше 40021 и больше 40119 приводит к ошибке адресации.

Таблица 13-147 Примеры для адресации регистра хранения Modbus

HR_Start_Offset	Адрес	Минимум	Максимум
0	Адрес Modbus (Word)	40001	40099
	Адрес S7-1200	MW100	MW298
20	Адрес Modbus (Word)	40021	40119
	Адрес S7-1200	MW100	MW298

HR_Start_Offset - значение типа word, которое определяет начальный адрес регистра хранения Modbus и сохраняется в экземплярном блоке данных MB_SERVER. Эти общедоступные статические переменные можно выбрать из выпадающего списка параметров, после того как в программу была вставлена инструкция MB_SERVER.

Например, после вставки MB_SERVER в сегмент LAD, можно перейти в предыдущий сегмент и назначить значение HR_Start_Offset. Значение необходимо назначить до исполнения MB_SERVER.



Ввод переменной для сервера Modbus через имя DB по умолчанию:

1. Установить курсор в поле параметра и ввести символ m.
2. В выпадающем списке имен DB выбрать "MB_SERVER_DB".
3. В выпадающем списке переменных DB выбрать "MB_SERVER_DB.HR_Start_Offset".

Таблица 13-148 Коды состояния выполнения MB_SERVER [1]

STATUS (W#16#....)	Код ответа an Modbus-Server (B#16#)	Ошибка протокола Modbus
7001		MB_SERVER ожидает, что клиент Modbus установит соединение с назначенным TCP портом. Этот код возвращается для первого выполнения операции подключения или отключения.
7002		MB_SERVER ожидает, что клиент Modbus установит соединение с назначенным TCP портом. Это будет сигнализировано для любого последующего выполнения при ожидании завершения операции подключения или отключения.
7003		Операция отключения успешно завершена (действует только для одного цикла PLC)
8187		Недействительный указатель на MB_HOLD_REG: Область слишком маленькая.
818C		Указатель на оптимизированною область MB_HOLD_REG (при этом речь должна идти об не оптимизированной области DB или области M памяти), или тайм-аут для заблокированного процесса превышает предельное значение в 55 секунд (спец. для S7-1200)
8381	01	Код функции не поддерживается
8382	03	Ошибка в длине данных
8383	02	Ошибка в адресе данных или доступ вне диапазона адресов MB_HOLD_REG
8384	03	Ошибка в значении данных
8385	03	Значение диагностического кода данных не поддерживается (код функции 08)

[1] В дополнение к упомянутым выше ошибкам MB_SERVER, ошибки могут быть возвращены из базовых коммуникационных инструкций T блока (TCON, TDISCON, TSEND и TRCV (Страница 864)).

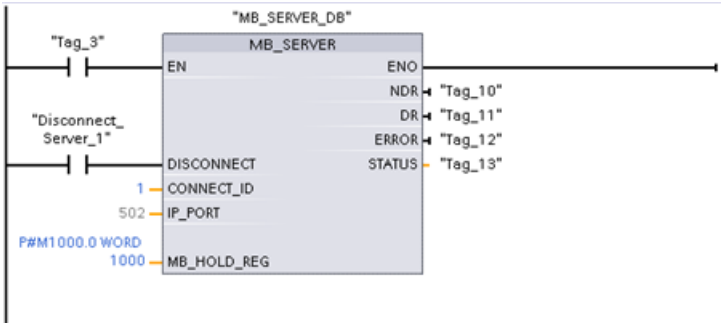
13.8.4 Старые примеры Modbus TCP

13.8.4.1 Пример: MB_SERVER для нескольких соединений TCP в старых системах

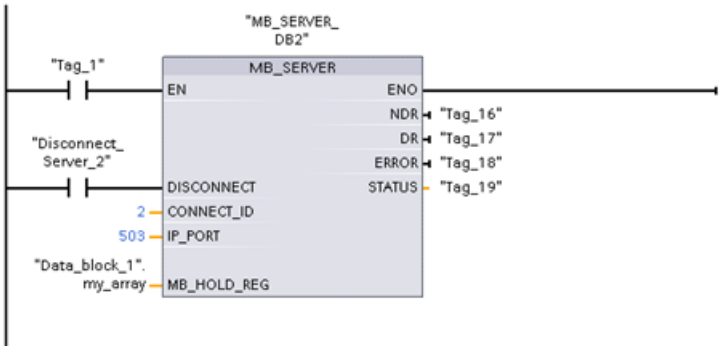
Возможно несколько соединений с сервером Modbus TCP. Для этого MB_SERVER должен выполняться независимо для каждого соединения. Каждое соединение должно использовать независимые экземплярный DB, ID соединения и IP порт. S7-1200 позволяет только одно соединение на IP-порт.

Для лучшей производительности MB_SERVER должен выполняться в каждом программном цикле для каждого соединения.

Сегмент 1: Соединение № 1 с независимыми IP_PORT, ID соединения и экземплярным DB



Сегмент 2: Соединение № 2 с независимыми IP_PORT, ID соединения и экземплярным DB



13.8.4.2 Пример: MB_CLIENT 1 в старых системах: Несколько запросов с общим TCP соединением

Несколько клиентских запросов Modbus могут быть отправлены по одному соединению. Для этого следует использовать тот же экземплярный DB , ID соединения и номер порта.

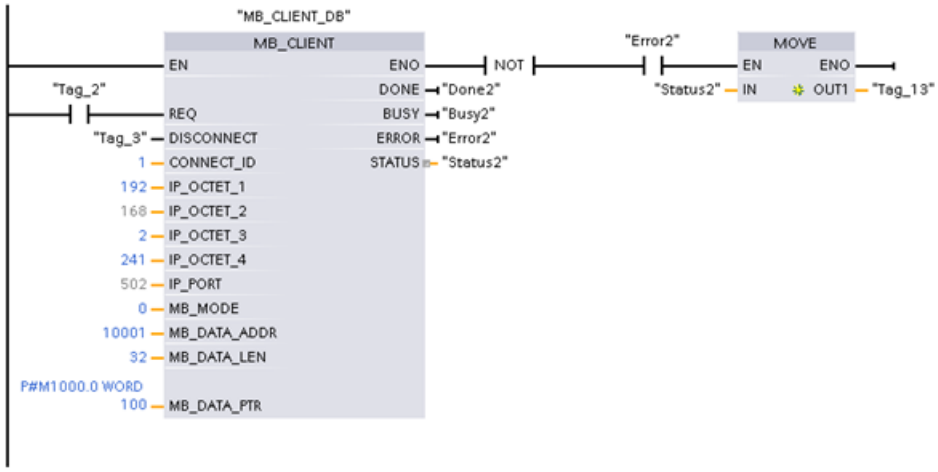
Только 1 клиент может быть активным в любой момент времени. Как только клиент завершает выполнение, следующий клиент может начать выполнение. Программа отвечает за последовательность выполнения.

Пример демонстрирует обоих клиентов, выполняющих запись в одну область памяти. Кроме того, возвращенная ошибка регистрируется, что является необязательным.

Сегмент 1: Modbus функция 1 - Читать 16 выходных битов образа процесса



Сегмент 2: Modbus функция 2 - Читать 32 входных бита образа процесса



13.8.4.3 Пример: MB_CLIENT 2 в старых системах: Несколько запросов с различными TCP соединениями

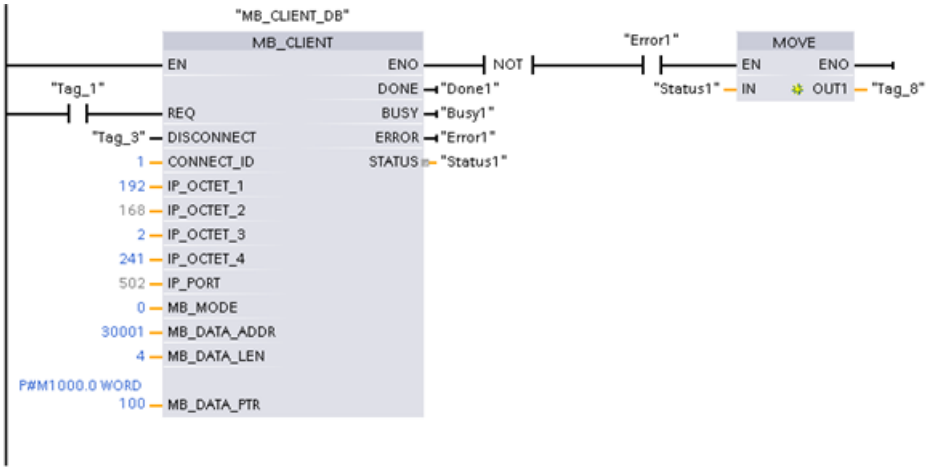
Несколько клиентских запросов Modbus могут быть отправлены через различные соединения. Для этого должны использоваться различные экземплярные DB, ID-адреса и ID соединения.

Номер порта должен отличаться, если соединения устанавливаются с тем же сервером Modbus. Если соединения находятся на различных серверах, нет никаких ограничений на номер порта.

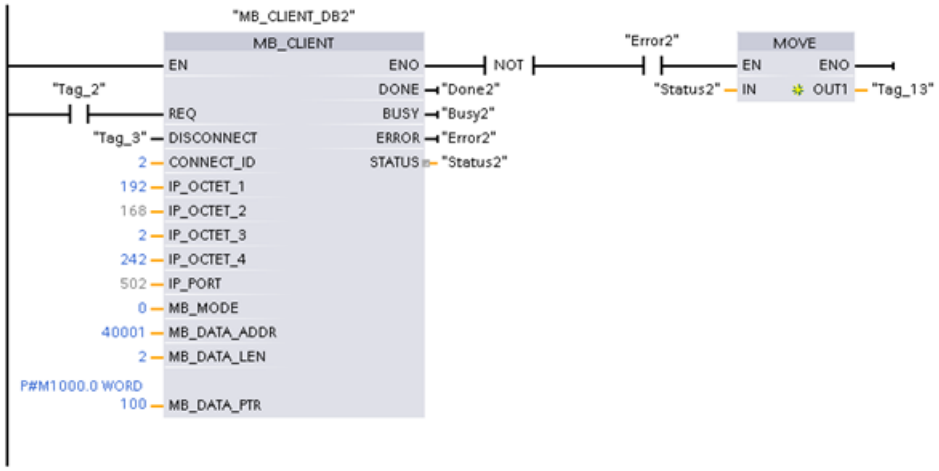
Пример демонстрирует обоих клиентов, выполняющих запись в одну область памяти. Кроме того, возвращенная ошибка регистрируется, что является необязательным.

Сегмент 1:

Функция Modbus 4 - Считывание входных слов (в памяти S7-1200)



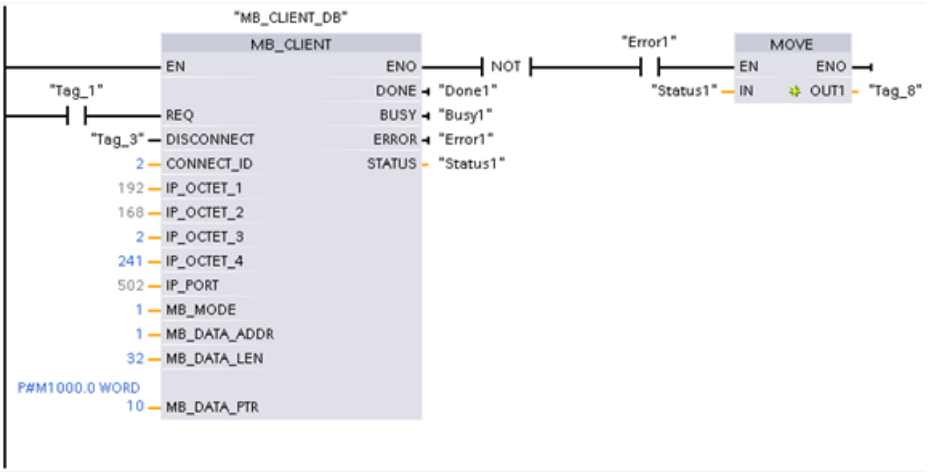
Сегмент 2: Modbus функция 3 – Считывание слов в регистре хранения на сервере Modbus TCP



13.8.4.4 Пример: MB_CLIENT 3 в старых системах: Запрос записи для образа процесса выходов

Этот пример показывает запрос клиента Modbus на запись в образ по выводу для S7-1200.

Сегмент 1: Modbus функция 15 – Записать биты в образ по выводу для S7-1200.

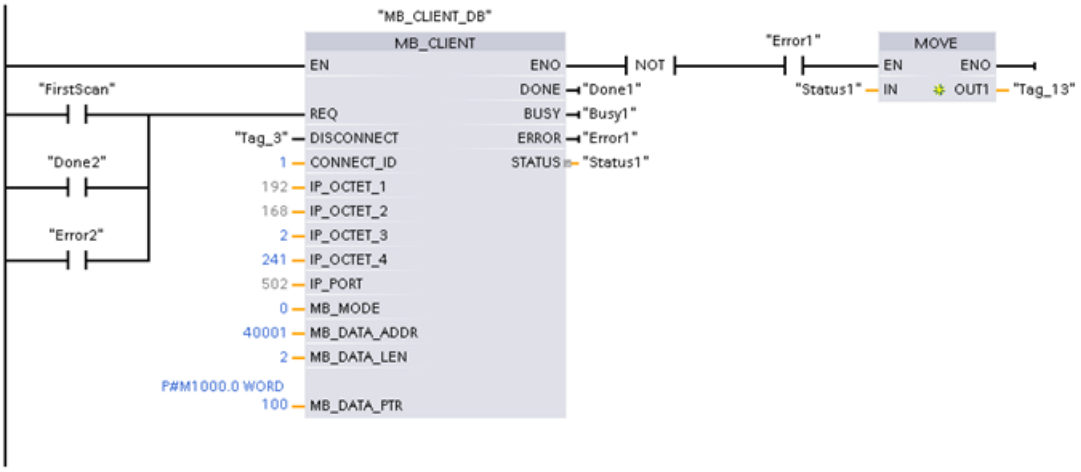


13.8.4.5 Пример: MB_CLIENT 4 в старых системах: Координация нескольких запросов

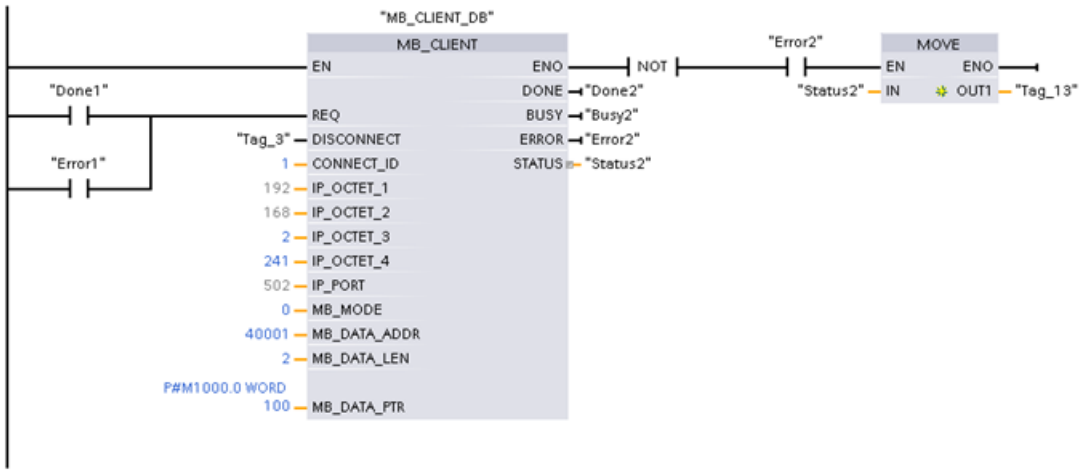
Необходимо обеспечить, чтобы выполнение каждого отдельного запроса Modbus TCP было бы завершено. Эту координацию должна выполнять программа. Приведенный ниже пример демонстрирует, как выходы первого и второго клиентских запросов могут координировать последовательностью выполнения.

Пример демонстрирует обоих клиентов, выполняющих запись в одну область памяти. Кроме того, возвращенная ошибка регистрируется, что является необязательным.

Сегмент 1: Modbus функция 3 – Считывание слов в регистре хранения



Сегмент 2: Modbus функция 3 – Считывание слов в регистре хранения



13.9 Modbus RTU коммуникация в старых системах (только CM/CB 1241)

13.9.1 Обзор

До STEP 7 версии V13 SP1 и CPU S7-1200 версии V4.1, Modbus RTU коммуникационные инструкции имели другие имена, и в некоторых случаях, несколько иные интерфейсы. Общие концепции применимы к обоим типам инструкций. См. описания отдельных Modbus RTU инструкций для получения информации о программировании.

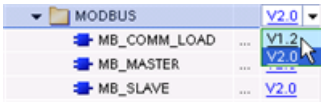
13.9.2 Выбор версии инструкций Modbus RTU

Существует две версии старых инструкций Modbus RTU в STEP 7:

- Старая версия 1.3: Совместима со всеми моделями CPU и CP
- Старая версия 2.2: Совместима со всеми моделями CPU и CP
  (Примечание: Версия 2.2 добавляет в MB_COMM_LOAD дополнительные параметры REQ и DONE. Теперь в V2.2 параметр MB_ADDR для MB_MASTER и MB_SLAVE разрешает UInt значение для расширенной адресации.)

Учитывая совместимость и для упрощения миграции можно выбирать, какая версия инструкций будет вставлена в программу пользователя.

Вызвать на карте задач "Инструкция" инструкции MODBUS в группе Коммуникационный процессор.



Для изменения версии Modbus инструкции, выбрать соответствующую версию в раскрывающемся списке. Можно выбрать группу или отдельные инструкции.

Если инструкция Modbus размещается через дерево инструкций в программе, то в дереве проекта создается новый экземпляр FB. Для просмотра нового экземпляра FB открыть в дереве проекта PLC_x > Программные блоки > Системные блоки > Программные ресурсы.

Чтобы узнать версию инструкции Modbus в программе, следует вызвать свойства в дереве проекта, а не свойства блочного элемента на экране в редакторе текстов программ. Выбрать в дереве проекта экземпляр Modbus FB, кликнуть правой кнопкой мыши, выбрать "Свойства" и после страницу "Информация", чтобы отобразить номер версии инструкции Modbus.

13.9.3 Старые инструкции Modbus RTU

13.9.3.1 MB_COMM_LOAD (конфигурирование порта на PtP модуле для Modbus RTU)

Таблица 13-149 Инструкция MB_COMM_LOAD

LAD/FBD	SCL	Описание

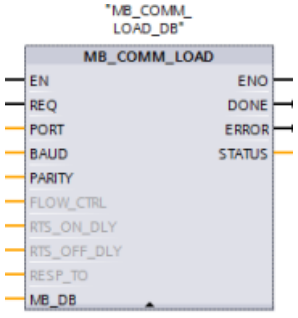
	`"MB_COMM_LOAD_DB"(`
	`REQ:=_bool_in,`
	`PORT:=_uint_in_,`
	`BAUD:=_udint_in_,`
	`PARITY:=_uint_in_,`
	`FLOW_CTRL:=_uint_in_,`
	`RTS_ON_DLY:=_uint_in_,`
	`RTS_OFF_DLY:=_uint_in_,`
	`RESP_TO:=_uint_in_,`
	`DONE=>_bool_out,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`MB_DB:=_fbtref_inout_);`
		Инструкция MB_COMM_LOAD конфигурирует PtP порт для коммуникации по протоколу Modbus-RTU. Аппаратные опции порта Modbus: Установка до трех CM (RS485 или RS232) плюс одна CB (R4845). При добавлении инструкции MB_COMM_LOAD в программу, автоматически назначается экземплярный блок данных.

Таблица 13-150 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	Инструкцию запускает положительный фронт (0 на 1). (только версия 2.0)
PORT	IN	Port	После установки и конфигурирования CM или CB, идентификатор порта появляется в выпадающем списке параметра на соединении блока PORT. Назначенное значение порта CM или CP является свойством "Аппаратный идентификатор" конфигурации устройства. Символьное имя порта назначается на вкладке "Системные константы" таблицы переменных PLC.
BAUD	IN	UDInt	Выбор скорости передачи данных: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200. Все остальные значения недопустимы.

Параметр и тип данных		Тип данных	Описание
PARITY	IN	UInt	Выбор четности: • 0 – отсутствует • 1 – совпадение при контроле по нечётности • 2 – совпадение при контроле по чётности
FLOW_CTRL [1]	IN	UInt	Выбор управления потоком: • 0 – (по умолчанию) без управления потоком • 1 – аппаратное управления потоком с "RTS всегда ВКЛ" (не действует для портов RS485) • 2 – аппаратное управления потоком с "RTS с переключением"
RTS_ON_DLY [1]	IN	UInt	Выбор задержки включения RTS: • 0 – (по умолчанию) без задержки от активного RTS до передачи первого символа сообщения • От 1 до 65535 – Задержка в миллисекундах от активного RTS до передачи первого символа сообщения (не применимо к портам RS485). Задержки RTS должны применяться независимо от выбора FLOW_CTRL.
RTS_OFF_DLY [1]	IN	UInt	Выбор задержки выключения RTS: • 0 – (по умолчанию) без задержки после передачи последнего символа и до "RTS неактивен" • От 1 до 65535 – задержка в миллисекундах от передачи последнего символа до "RTS неактивен" (не действует для портов RS485). Задержки RTS должны применяться независимо от выбора FLOW_CTRL.
RESP_TO [1]	IN	UInt	Тайм-аут ответа: Время в миллисекундах, в течение которого MB_MASTER ждет ответа от ведомого устройства. Если ведомое устройство не отвечает в течение этого времени, MB_MASTER повторяет запрос или завершает запрос с ошибкой, если выполнено заданное количество повторов. От 5 до 65 535 мс (значение по умолчанию = 1000 мс).
MB_DB	IN	Variant	Ссылка на экземплярный блок данных инструкций MB_MASTER или MB_SLAVE. После вставки MB_SLAVE или MB_MASTER в программу, идентификатор DB будет доступен в выпадающем списке параметра на соединении блока MB_DB.
DONE	OUT	Bool	Бит DONE = TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок. (только версия 2.0)
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение TRUE, после того, как последний запрос был завершен с ошибкой. Код ошибки в параметре STATUS действителен только в цикле, где ERROR = TRUE.
STATUS	OUT	Word	Условие выполнения

[1] Опциональные параметры для MB_COMM_LOAD (V 2.x или выше). Кликнуть по стрелке у основания блока LAD/FBD, чтобы развернуть его и отобразить эти параметры.

MB_COMM_LOAD выполняется, чтобы выполнить конфигурацию порта для протокола Modbus-RTU. После того как была выполнена конфигурация порта для протокола Modbus RTU, он может быть использован инструкциями MB_MASTER или MB_SLAVE.

Для конфигурации каждого коммуникационного порта, который используется для коммуникации Modbus, необходимо исполнить MB_COMM_LOAD. Каждому

используемому порту необходимо присвоить уникальный экземплярный блок данных MB_COMM_LOAD. Можно установить до трех коммуникационных модулей (RS232 или RS485) и одну коммуникационную плату (RS485) в CPU. Можно вызвать MB_COMM_LOAD из пускового OB и выполнить ее однократно или можно использовать системный флаг первого цикла (Страница 109), чтобы инициировать однократное выполнение. Повторное выполнение MB_COMM_LOAD требуется только тогда, если необходимо изменить параметры передачи данных, например скорость передачи данных или четность.

При добавлении MB_MASTER или MB_SLAVE в программу, инструкции назначается экземплярный блок данных. На этот экземплярный блок данных делается ссылка при указании параметра MB_DB в инструкции MB_COMM_LOAD.

Переменные блока данных MB_COMM_LOAD

В таблице ниже приведены общедоступные статические переменные в экземплярном блоке данных MB_COMM_LOAD, которые могут быть использованы в программе.

Таблица 13-151 Статические переменные экземплярного блока данных

Переменная	Тип данных	Описание
ICHAR_GAP	UInt	Задержка расстояния между символами. Данный параметр указывается в миллисекундах и предназначен для увеличения ожидаемого периода времени между принимаемыми символами. Соответствующее количество битовых интервалов для этого параметра добавляется к стандартному значению Modbus в 35 битовых интервалов (3,5 x время символа).
RETRIES	UInt	Количество повторных попыток, которые выполняет ведущее устройство, прежде чем будет возвращен кода ошибки 0x80C8 для "Ответ отсутствует".
STOP_BITS	USInt	Количество стоповых битов, используемых в синхронизации каждого символа. Допустимыми значениями являются 1 и 2.

Таблица 13-152 Коды состояния выполнения MB_COMM_LOAD [1]

STATUS (W#16#....)	Описание
0000	Ошибки отсутствуют
8180	Недопустимое значение ID порта (неправильный идентификатор порта/оборудования у коммуникационного модуля)
8181	Недопустимое значение для скорости передачи данных.
8182	Недопустимое значение для четности.
8183	Недопустимое значение для управления потоком.
8184	Недопустимое значение для тайм-аута ответа (тайм-аут ответа меньше, чем минимальное значение в 5 мс)
8185	Параметр MB_DB не является экземплярным блоком данных для инструкции MB_MASTER или MB_SLAVE.

[1] Наряду с перечисленными выше ошибками инструкций MB_COMM_LOAD, базовые PtP коммуникационные инструкции также могут возвращать ошибки.

13.9.3.2 MB_MASTER (обмен данными через PtP-порт в качестве ведущего устройства Modbus RTU)

Таблица 13-153 Инструкция MB_MASTER

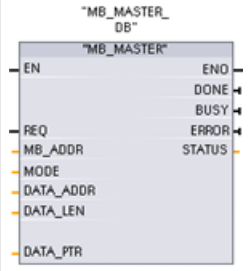
LAD/FBD	SCL	Описание
	`"MB_MASTER_DB"( REQ:=_bool_in_, MB_ADDR:=_uint_in_, MODE:=_usint_in_, DATA_ADDR:=_udint_in_, DATA_LEN:=_uint_in_, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, DATA_PTR:=_variant_inout_);`	Инструкция MB_MASTER выполняет коммуникацию как ведущее устройство Modbus через порт, конфигурация которого была при прошлом вызове инструкции MB_COMM_LOAD. При добавлении инструкции MB_MASTER в программу, автоматически назначается экземплярный блок данных. Этот экземплярный блок данных MB_MASTER используется при указании параметра MB_DB в инструкции MB_COMM_LOAD.

Таблица 13-154 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
REQ	IN	Bool	0 = запрос отсутствует 1 = запрос для отправки данных на ведомое устройство Modbus
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Адрес станции Modbus-RTU: Стандартный диапазон адресации (от 1 до 247) Расширенный диапазон адресации (от 1 до 65535) Значение "0" зарезервировано для рассылки сообщения на все ведомые устройства Modbus. Только коды функций Modbus 05, 06, 15 и 16 поддерживаются широковещательной передачей.
MODE	IN	USInt	Выбор режима: Определяет тип запроса (чтение, запись или диагностика). Дополнительную информацию можно найти в таблице функций Modbus ниже.
DATA_ADDR	IN	UDInt	Начальный адрес в ведомом устройстве: Задает начальный адрес данных, к которым будет выполняться доступ в ведомом устройстве Modbus. Действительные адреса можно найти в таблице функций Modbus ниже.
DATA_LEN	IN	UInt	Длина данных: Указывает количество битов или слов, к которым будет обращаться этот запрос. Действительные значения длины можно найти в таблице функций Modbus ниже.
DATA_PTR	IN	Variant	Указатель данных: Указывает на M или DB адрес (не оптимизированный DB) для записываемых или считываемых данных.
DONE	OUT	Bool	Бит DONE = TRUE в течение одного цикла, после того, как последний запрос был выполнен без ошибок.
BUSY	OUT	Bool	• 0 - нет обрабатываемых инструкций MB_MASTER • 1 - обрабатывается инструкция MB_MASTER

Параметр и тип данных		Тип данных	Описание
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение TRUE, после того, как последний запрос был завершен с ошибкой. Код ошибки в параметре STATUS действителен только в цикле, где ERROR = TRUE.
STATUS	OUT	Word	Условие выполнения

Правила для коммуникации ведущего устройства Modbus

- MB_COMM_LOAD должна выполняться для конфигурирования порта, чтобы инструкция MB_MASTER могла связываться с этим портом.
- Если порт должен использоваться, чтобы инициировать запросы ведущего устройства Modbus, то порт не должен использоваться MB_SLAVE. Один или несколько экземпляров MB_MASTER могут использоваться с этим портом. Однако для исполнения всех MB_MASTER необходимо использовать одинаковый экземплярный блок данных MB_MASTER для порта.
- Инструкции Modbus не используют коммуникационные прерывающие события для управления процессом коммуникации. Программа должна опрашивать инструкцию MB_MASTER на наличие завершенных процессов отправки и получения.
- Рекомендуется вызывать MB_MASTER для всех исполнений для определенного порта из OB цикла программы. Инструкции ведущего устройства Modbus могут выполняться только в одном цикле программы соответственно или на циклических/с временной задержкой уровнях выполнения. Они не должны выполняться на обоих уровнях приоритета выполнения. Прерывание по приоритету инструкции ведущего устройства Modbus другой инструкцией ведущего устройства Modbus на уровне выполнения с более высоким приоритетом приводит к нештатной работе. Инструкции ведущего устройства Modbus не должны обрабатываться на уровнях выполнения, связанных с запуском, диагностикой или ошибкой времени выполнения.
- Как только инструкция ведущего устройства инициирует передачу, этот экземпляр должен постоянно выполняться с активным входом EN, до возврата состояния DONE =1 или ERROR =1 . Определенный экземпляр MB_MASTER считают активным, пока не наступает одно из этих двух событий. В то время как исходный экземпляр активен, любой вызов другого экземпляра с активным входом REQ приведет к ошибке. Если непрерывное выполнение исходного экземпляра останавливается, состояние запроса остается активным на время, определенное статической переменной Blocked_Proc_Timeout. Как только этот время истекает, следующая вызванная инструкция ведущего устройства с активным входом REQ станет активным экземпляром. Это препятствует тому, чтобы единственная инструкция ведущего устройства Modbus монополизировала или заблокировал доступ к порту. Если исходный активный экземпляр не будет активирован в течение времени, определенного статической переменной "Blocked_Proc_Timeout", то следующее выполнение этого экземпляра (без установленного входа REQ) сбросит активное состояние. Если установлен, то это выполнение инициирует новый запрос ведущего устройства, как будто никакой другой экземпляр не был активен.

Параметр REQ

0 = запрос отсутствует; 1 = запрос для отправки данных на ведомое устройство Modbus

Можно управлять этим входом с помощью активируемого уровнем или фронтом контакта. Каждый раз, когда этот вход активирован, запускается конечный автомат, чтобы гарантировать, что другая инструкция MB_MASTER, использующий тот же экземплярный DB, не выставить запрос, пока текущий запрос не завершен. Состояния всех других входов регистрируются и сохраняются в системе для текущего запроса, пока не будет получен ответ или не будет обнаружена ошибка.

Если тот же экземпляр MB_MASTER будет выполняться снова с входом REQ =1 до завершения текущего запроса, то последующие передачи не будут выполнены. Однако, когда запрос завершен, новый запрос выставляется каждый раз, когда MB_MASTER выполняется вновь с входом REQ = 1.

Через параметры DATA_ADDR и MODE выбирается тип функции Modbus.

DATA_ADDR (начальный адрес Modbus в ведомом устройстве): Задает начальный адрес данных, к которым будет выполняться доступ в ведомом устройстве Modbus.

Инструкция MB_MASTER использует вместо входа кода функции вход MODE. Комбинация из MODE и адреса Modbus определяет код функции, используемый в собственном сообщении Modbus. В нижеследующей таблице показано отношение между параметром MODE, кодом функции Modbus и диапазоном адресов Modbus.

Таблица 13-155 Функции Modbus

MODE	Функция Modbus	Длина данных	Операция и данные	Адрес Modbus
0	01	От 1 до 2000 От 1 до 1992 [1]	Чтение выходных битов: От 1 до (1992 или 2000) бит за запрос	От 1 до 9999
0	02	От 1 до 2000 От 1 до 1992 [1]	Чтение входных битов: От 1 до (1992 или 2000) бит за запрос	От 10001 до 19999
0	03	От 1 до 125 От 1 до 124 [1]	Чтение регистра хранения: От 1 до (124 или 125) слов за запрос	От 40001 до 49999 или от 400001 до 465535
0	04	От 1 до 125 От 1 до 124 [1]	Чтение входных слов: От 1 до (124 или 125) слов за запрос	От 30001 до 39999
1	05	1	Запись выходного бита: Один бит на запрос	От 1 до 9999
1	06	1	Запись регистра хранения: 1 слово на запрос	От 40001 до 49999 или от 400001 до 465535
1	15	От 2 до 1968 От 2 до 1960 [1]	Запись нескольких выходных битов: От 2 до (1960 или 1968) бит за запрос	От 1 до 9999
1	16	От 2 до 123 От 2 до 122 [1]	Запись нескольких регистров хранения: От 2 до (122 или 123) слов за запрос	От 40001 до 49999 или от 400001 до 465535
2	15	От 1 до 1968 От 2 до 1960 [1]	Запись одного или нескольких выходных битов: От 1 до (1960 или 1968) бит за запрос	От 1 до 9999
2	16	От 1 до 123 От 1 до 122 [1]	Запись одного или нескольких регистров хранения: От 1 до (122 или 123) слов за запрос	От 40001 до 49999 или от 400001 до 465535

MODE	Функция Modbus	Длина данных	Операция и данные	Адрес Modbus
11	11	0	Считать слово состояния и счетчик событий коммуникации ведомого устройства. Слово состояния указывает на "занято" (0 – не занято, 0xFFFF – занято). Счетчик событий инкрементируется при каждой удачной обработке сообщения. Оба операнда DATA_ADDR b DATA_LEN из MB_MASTER игнорируются данной функцией.	
80	08	1	Проверка состояния ведомого устройства диагностическим кодом данных 0x0000 (петлевой тест – ведомое устройство возвращает эхо запроса) 1 слово на запрос	
81	08	1	Сброс счетчика событий ведомого устройства посредством диагностического кода данных 0x000A 1 слово на запрос	
3 до 10, 12 до 79, 82 до 255			Зарезервировано	

[1] При расширенной адресации максимальная длина данных, в зависимости от типа данных функции, меньше на 1 байт или 1 слово.

Параметр DATA_PTR

Параметр DATA_PTR указывает на адрес в DB или M области, в который выполняется записи или из которого выполняется чтение. При использовании блока данных необходимо создать глобальный блок данных, который предоставит память данных для операций чтения и записи на ведомые устройства Modbus.

Примечание

Блок данных DATA_PTR должен допускать прямую адресацию.

Блок данных должен допускать прямую (абсолютную) и символьную адресацию. При создании блока данных следует выбрать атрибут доступа "Стандарт".

Структуры блока данных для параметра DATA_PTR

- Эти типы данных действуют для **считывания слов** адресов Modbus от 30001 до 39999, от 40001 до 49999 и от 400001 до 465536, а также для **записи слов** в адреса Modbus от 40001 до 49999 и от 400001 до 465536.
  - Стандартный массив типов данных WORD, UINT или INT.
  - Именованная структура типа WORD, UINT или INT, где каждый элемент имеет уникальное имя и 16-битный тип данных.
  - Именованная сложная структура, где каждый элемент имеет уникальное имя и 16- или 32-битный тип данных.
- Для **чтения** и записи битов для адресов Modbus с 00001 по 09999 и для чтения битов с 10001 по 19999.
  - Стандартный массив булевых типов данных.
  - Именованная булева структура уникально именованных булевых переменных.
- Наличие отдельной области памяти для каждой инструкции MB_MASTER не обязательно, но рекомендуется. Причина этой рекомендации кроется в том, что есть большая вероятность повреждения данных, если несколько инструкций MB_MASTER читают и пишут в ту же самую область памяти.
- Не обязательно, чтобы области данных для DATA_PTR находились в одном и том же глобальном блоке данных. Можно создать один блок данных с несколькими областями для процессов считывания Modbus, один блок данных для процессов записи Modbus или один блок данных для каждой станции ведомого устройства.

Переменные в блоке данных ведущего устройства Modbus

В таблице ниже приведены общедоступные статические переменные в экземплярном блоке данных MB_MASTER, которые могут быть использованы в программе.

Таблица 13-156 Статические переменные экземплярного блока данных

Переменная	Тип данных	Начальное значение	Описание
Blocked_Proc_Timeout	Real	3.0	Время (в секундах), в течение которого ожидается заблокированный экземпляр ведущего устройства Modbus, прежде чем этот экземпляр будет удален со статусом ACTIVE. Такое, например, возможно, если выдается запрос ведущего устройства и программа прекращает вызов функции ведущего устройства, прежде чем запрос будет полностью завершен. Значение времени должно быть больше 0 и меньше 55 секунд. В противном случае появится ошибка. Значение по умолчанию составляет 5 секунд.
Extended_Addressing	Bool	FALSE	Конфигурирует одно- или двухбайтовую адресацию ведомой станции. Стандартное значение равно 0. (0 = однобайтовый адрес, 1 = двухбайтовый адрес)

Программа может записывать значения в переменные Blocked_Proc_Timeout и Extended_Addressing , чтобы управлять операциями ведущего устройства Modbus. В разделе, относящемся к инструкции MB_SLAVE, в описании переменных HR_Start_Offset и Extended_Addressing можно найти пример того, как использовать эти переменные в

редакторе текстов программ, а также другая подробная информация о расширенной адресации Modbus (Страница 1356).

Коды условий

Таблица 13-157 Коды условий для выполнения MB_MASTER (ошибки коммуникации и конфигурации) [1]

STATUS (W#16#....)	Описание
0000	Ошибки отсутствуют
80C8	Тайм-аут ведомого устройства. Проверить скорость передачи данных, четность и подключения ведомого устройства.
80D1	Получатель запросил управление потоком для прерывания текущей передачи, но не активировал передачу снова в течение определенного времени ожидания. Эта ошибка также генерируется при аппаратном управлении потоком, если получатель в течение времени ожидания не распознает CTS.
80D2	Запрос на передачу был прерван, так как сигнал DSR не получен от DCE.
80E0	Сообщение завершено, так как принимающий буфер заполнен.
80E1	Сообщение завершено из-за ошибки четности.
80E2	Сообщение завершено из-за ошибки телеграммы.
80E3	Сообщение завершено из-за ошибки переполнения.
80E4	Сообщение отменено, так как указанная длина превышает суммарный размер буфера.
8180	Недопустимое значение для ID порта или ошибка инструкции MB_COMM_LOAD
8186	Недействительный адрес станции Modbus
8188	Недопустимый режим определен для широковещательного запроса
8189	Недопустимое значение для адреса данных.
818A	Недопустимое значение для длины данных.
818B	Недействительный указатель на локальный источник данных, т. е. на локальную цель данных: Неверный размер
818C	Недопустимый указатель для DATA_PTR или недопустимый Blocked_Proc_Timeout: Область данных должна быть представлена DB (который позволяет как символьную, так и прямую адресацию) или M памятью
8200	Порт занят обработкой запроса передачи.

Таблица 13-158 Коды условий для выполнения MB_MASTER (ошибка протокола Modbus) [1]

STATUS (W#16#....)	Код ответа ведомого устройства	Ошибка протокола Modbus
8380	-	Ошибка контр. суммы
8381	01	Код функции не поддерживается
8382	03	Ошибка в длине данных
8383	02	Ошибка в адресе данных или адрес вне действительного диапазона DATA_PTR
8384	Больше чем 03	Ошибка в значении данных

STATUS (W#16#....)	Код ответа ведомого устройства	Ошибка протокола Modbus
8385	03	Значение диагностического кода данных не поддерживается (код функции 08)
8386	-	Код функции в ответе не соответствует коду в запросе.
8387	-	Ответило неправильное ведомое устройство.
8388	-	Ответ ведомого устройства на запрос записи неправильный. Запрос записи, возвращенный ведомым устройством, не соответствует отправленному ведущим устройством.

[1] Наряду с перечисленными выше ошибками инструкции MB_MASTER, базовые PtP коммуникационные инструкции также могут возвращать ошибки.

13.9.3.3 MB_SLAVE (обмен данными через PtP-порт в качестве ведомого устройства Modbus RTU)

Таблица 13-159 Инструкция MB_SLAVE

LAD/FBD	SCL	Описание

	`"MB_SLAVE_DB"(`
	`MB_ADDR:=_uint_in_,`
	`NDR=>_bool_out_,`
	`DR=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`MB_HOLD_REG:=_variant_inout_);`
		При помощи инструкции MB_SLAVE программа может выполнять коммуникацию в качестве ведомого устройства через PtP порт на CM (RS485 или RS232) или CB (RS485). Когда удаленное ведущее устройство Modbus RTU выставляет запрос, пользовательская программа отвечает на запрос выполнением инструкции MB_SLAVE. При добавлении инструкции, STEP 7 автоматически создает DB экземпляра. С помощью этого имени MB_SLAVE_DB задается параметр MB_DB для инструкции MB_COMM_LOAD.

Таблица 13-160 Типы данных для параметров

Параметр и тип данных		Тип данных	Описание
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Адрес станции ведомого устройства Modbus: Стандартный диапазон адресации (от 1 до 247) Расширенный диапазон адресации (от 0 до 65535)
MB_HOLD_REG	IN	Variant	Указатель на DB регистра хранения Modbus: регистр хранения Modbus может быть M памятью или блоком данных.
NDR	OUT	Bool	Новые данные готовы: • 0 – Новых данных нет • 1 – Указывает, что новые данные были записаны ведущим устройством Modbus

Параметр и тип данных		Тип данных	Описание
DR	OUT	Bool	Считывание данных: • 0 – Данные не считаны • 1 – Указывает, что данные были считаны ведущим устройством Modbus
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение TRUE, после того, как последний запрос был завершен с ошибкой. Если исполнение завершается ошибкой, код ошибки в параметре STATUS действителен только на протяжении цикла, в котором ERROR = TRUE.
STATUS	OUT	Word	Код ошибки выполнения

Коды функций коммуникации Modbus (1, 2, 4, 5 и 15) могут считывать и записывать биты и слова непосредственно в образе процессов входов и в образе процессов выходов CPU. Для этих функциональных кодов параметр MB_HOLD_REG в качестве типа данных должен быть больше байта. Следующая таблица отражает примерное присвоение адресов Modbus для образа процесса в CPU.

Таблица 13-161 Согласование адресов Modbus для образа процесса

Функции Modbus						S7-1200	
Коды	Функция	Область данных	Диапазон адресов			Область данных	Адрес CPU
01	Читать биты	Вывод	1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7
02	Читать биты	Ввод	10001	до	18192	Образ процесса по вводу	От I0.0 до I1023.7
04	Чтение слов	Ввод	30001	до	30512	Образ процесса по вводу	От IW0 до IW1022
05	Записать бит	Вывод	1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7
15	Записать биты	Вывод	1	до	8192	Образ процесса по выводу	От Q0.0 до Q1023.7

Функциональные коды коммуникации Modbus (3, 6, 16) используют регистр хранения, под которым может подразумеваться диапазон адресов в M памяти или блок данных. Тип регистра хранения указывает параметр MB_HOLD_REG инструкции MB_SLAVE.

Примечание

Тип блока данных MB_HOLD_REG

Блок данных с регистром хранения Modbus должен разрешать прямую (абсолютную) и символьную адресацию. При создании блока данных следует выбрать атрибут доступа "Стандарт".

Следующая таблица показывает примеры согласования адресов Modbus с регистрами хранения, используемые для кодов функций Modbus 03 (чтение слов), 06 (запись слов)

и 16 (запись слов). Фактический верхний предел адресов DB определен максимальным объемом рабочей памяти и объемом M памяти для каждой из моделей CPU.

Таблица 13-162 Согласование адресов Modbus для памяти CPU

Адрес ведущего устройства Modbus	**Примеры для параметра MB_HOLD_REG**				
	MW100	**DB10.DBW0**	**MW120**	**DB10.DBW50**	**"Recipe".ingredient**
40001	MW100	DB10.DBW0	MW120	DB10.DBW50	"Recipe".ingredient[1]
40002	MW102	DB10.DBW2	MW122	DB10.DBW52	"Recipe".ingredient[2]
40003	MW104	DB10.DBW4	MW124	DB10.DBW54	"Recipe".ingredient[3]
40004	MW106	DB10.DBW6	MW126	DB10.DBW56	"Recipe".ingredient[4]
40005	MW108	DB10.DBW8	MW128	DB10.DBW58	"Recipe".ingredient[5]

Таблица 13-163 Функции диагностики

Функции диагностики MB_SLAVE для S7-1200		
Коды	Подфункция	Описание
08	0000H	Вывод данных запроса эхо-контроля: Инструкция MB_SLAVE возвращает ведущему устройству Modbus эхо полученного слова данных.
08	000AH	Очистка счетчика событий коммуникации: Инструкция MB_SLAVE очищает счетчик событий коммуникации, который используется для функции Modbus 11.
11		Вызов счетчика событий коммуникации: Инструкция MB_SLAVE использует внутренний счетчик событий коммуникации, чтобы определять количество выполненных запросов чтения и записи Modbus, которые отправляются на ведомое устройство Modbus. Значение счетчика не увеличивается при выполнении функций 8, функции 11 и запросов широкого вещания. Кроме этого, значение счетчика не увеличивается при запросах, которые приводят к ошибкам коммуникации (например, ошибка четности или контрольной суммы).

Инструкция MB_SLAVE поддерживает широковещательные запросы записи от любого ведущего устройства Modbus, если запрос использует допустимые адреса. MB_SLAVE создает код ошибки "0x8188" для кодов функций, не поддерживаемых в широковещательной передаче.

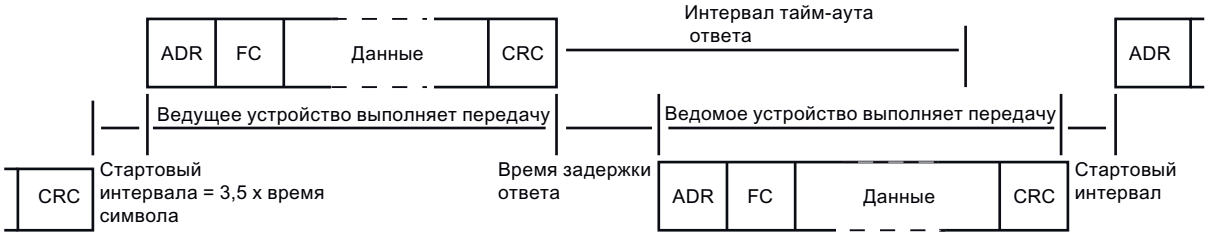
Правила для коммуникации ведомого устройства Modbus

- MB_COMM_LOAD должна выполняться для конфигурирования порта, чтобы инструкция MB_SLAVE могла выполнить коммуникацию через этот порт.
- Если порт должен реагировать на ведущее устройство Modbus как ведомое устройство, программирование этого порта при помощи инструкции MB_MASTER запрещено.
- С определенным портом может использоваться только один экземпляр MB_SLAVE, в противном случае возможно неожиданное поведение.

- Инструкции Modbus не используют коммуникационные прерывающие события для управления процессом коммуникации. Программа должна считывать коммуникационный процесс путем опроса инструкции MB_SLAVE на наличие завершенных процессов отправки и получения.
- Инструкция MB_SLAVE должна выполняться регулярно с частотой, которая позволяет давать в указанное время ответ на входящие запросы ведущего устройства Modbus. Рекомендуется выполнять MB_SLAVE в каждом цикле из OB цикла программы. MB_SLAVE может выполняться из OB циклического прерывания. Однако это не рекомендуется, так как из-за слишком больших задержек по времени в прерывающей программе выполнение других программ прерывания может быть временно блокировано.

Временная диаграмма сигнала Modbus

MB_SLAVE должна выполняться регулярно, чтобы принимать каждый запрос ведущего устройства Modbus и отвечать на него соответствующим образом. Частота выполнения MB_SLAVE зависит от значения тайм-аута ответа, заданного ведущим устройством Modbus. Это показано на следующем рисунке.



Период тайм-аута для ответа RESP_TO - это промежуток времени, в течение которой ведущее устройство Modbus ожидает начала ответа от ведомого устройства Modbus. Этот период задается не протоколом Modbus, а параметром соответствующего ведущего устройства Modbus. Частота выполнения (время между выполнением и следующим выполнением) инструкции MB_SLAVE задается в зависимости от параметров ведущего устройства Modbus. Инструкцию MB_SLAVE необходимо выполнять как минимум два раза во время периода тайм-аута для ответа ведущего устройства Modbus.

Переменные ведомого устройства Modbus

В таблице ниже приведены общедоступные статические переменные в экземплярном блоке данных MB_SLAVE, которые могут быть использованы в программе.

Таблица 13-164 Переменные ведомого устройства Modbus

Переменная	Тип данных	Описание
Request_Count	Word	Общее количество запросов, полученных этим ведомым устройством
Slave_Message_Count	Word	Количество запросов, полученных для этого конкретного ведомого устройства
Bad_CRC_Count	Word	Количество полученных запросов, содержащих ошибку контрольной суммы
Broadcast_Count	Word	Количество полученных широковещательных запросов

Переменная	Тип данных	Описание
Exception_Count	Word	Конкретные ошибки Modbus, требующие возвращенного исключения
Success_Count	Word	Количество полученных для этого конкретного ведомого устройства запросов без ошибок протокола
HR_Start_Offset	Word	Задает начальный адрес регистра хранения Modbus (по умолчанию = 0)
Extended_Addressing	Bool	Конфигурирует одно- или двухбайтовую адресацию ведомого устройства (0 = однобайтовый адрес, 1 = двухбайтовый адрес, значение по умолчанию = 0)

Программа может записывать значения в переменные HR_Start_Offset и Extended_Addressing и управлять операциями ведомого устройства Modbus. Другие переменные могут считываться для контроля состояния Modbus.

HR_Start_Offset

Адреса регистра хранения Modbus начинаются с 40001 или 400001. Эти адреса соответствуют начальному адресу регистра хранения в памяти целевой системы. Однако можно использовать переменную HR_Start_Offset, чтобы сконфигурировать начальный адрес для регистра хранения Modbus, отличный от 40001 или 400001.

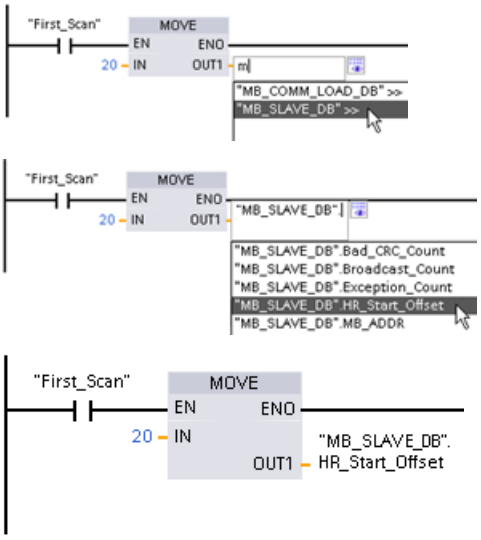
Можно, например, сконфигурировать регистр хранения с началом в MW100 и длиной в 100 слов. При помощи смещения 20 можно задать начальный адрес регистра хранения 40021 вместо 40001. Любой адрес меньше 40021 и больше 400119 приводит к ошибке адресации.

Таблица 13-165 Примеры для адресации регистра хранения Modbus

HR_Start_Offset	Адрес	Минимум	Максимум
0	Адрес Modbus (Word)	40001	40099
	Адрес S7-1200	MW100	MW298
20	Адрес Modbus (Word)	40021	40119
	Адрес S7-1200	MW100	MW298

HR_Start_Offset - значение типа word, которое определяет начальный адрес регистра хранения Modbus и сохраняется в экземплярном блоке данных MB_SLAVE. Эти общедоступные статические переменные можно выбрать из выпадающего списка параметров, после того как в программу была вставлена инструкция MB_SLAVE.

Например, после вставки MB_SLAVE в сегмент LAD, можно перейти в предыдущий сегмент и назначить значение HR_Start_Offset. Значение необходимо назначить до исполнения MB_SLAVE.



Ввод переменной ведомого устройства Modbus через имя блока данных по умолчанию:

1. Установить курсор в поле параметра и ввести символ m.
2. В выпадающем списке выбрать "MB_SLAVE_DB".
3. Установить курсор справа от имени блока данных (после кавычек) и ввести точку.
4. В выпадающем списке выбрать "MB_SLAVE_DB.HR_Start_Offset".

Extended_Addressing

Доступ к переменной Extended_Addressing осуществляется аналогичным способом, как и к представленной выше ссылке HR_Start_Offset, за исключением того случая, когда переменная Extended_Addressing является булевым значением. Булева переменная должна быть записана с помощью выходной катушки, а не блока перемещения.

Адресация ведомого устройства Modbus может быть сконфигурирована либо как однобайтовая (что является стандартом Modbus), либо, как двухбайтовая. Расширенная адресация используется для адресации более 247 устройств в одной сети. Выбор расширенной адресации позволяет адресовать максимум 64.000 адресов. В дальнейшем в качестве примера будет рассмотрена телеграмма функции 1 Modbus.

Таблица 13-166 Однобайтовый адрес ведомого устройства (байт 0)

Функция 1	**Байт 0**	**Байт 1**	**Байт 2**	**Байт 3**	**Байт 4**	**Байт 5**	
Запрос	Адрес ведомого устройства	Код функции	Начальный адрес		Длина катушек		
Действительный ответ	Адрес ведомого устройства	Код функции	Длина	Данные катушек			
Ответ с ошибкой	Адрес ведомого устройства	0x81	Код ошибки				

Таблица 13-167 Двухбайтовый адрес ведомого устройства (байты 0 и 1)

	Байт 0	**Байт 1**	**Байт 2**	**Байт 3**	**Байт 4**	**Байт 5**	**Байт 6**
Запрос	Адрес ведомого устройства		Код функции	Начальный адрес		Длина катушек	
Действительный ответ	Адрес ведомого устройства		Код функции	Длина	Данные катушек		
Ответ с ошибкой	Адрес ведомого устройства		0x81	Код ошибки			

Коды условий

Таблица 13-168 Коды условий для выполнения MB_SLAVE (ошибки коммуникации и конфигурации) [1]

STATUS (W#16#....)	**Описание**
80D1	Получатель запросил управление потоком для прерывания текущей передачи, но не активировал передачу снова в течение определенного времени ожидания. Эта ошибка также генерируется при аппаратном управлении потоком, если получатель в течение времени ожидания не распознает CTS.
80D2	Запрос на передачу был прерван, так как сигнал DSR не получен от DCE.
80E0	Сообщение завершено, так как принимающий буфер заполнен.
80E1	Сообщение завершено из-за ошибки четности.
80E2	Сообщение завершено из-за ошибки телеграммы.
80E3	Сообщение завершено из-за ошибки переполнения.
80E4	Сообщение отменено, так как указанная длина превышает суммарный размер буфера.
8180	Недопустимое значение для ID порта или ошибка инструкции MB_COMM_LOAD
8186	Недействительный адрес станции Modbus
8187	Недействительный указатель на MB_HOLD_REG-DB: Область слишком маленькая.
818C	Недопустимый указатель MB_HOLD_REG на M память или DB (область DB должна допускать, как символьную, так и абсолютную адресацию)

Таблица 13-169 Коды условий для выполнения MB_SLAVE (ошибка протокола Modbus) [1]

STATUS (W#16#....)	**Код ответа ведомого устройства**	**Ошибка протокола Modbus**
8380	Нет ответа	Ошибка контр. суммы
8381	01	Код функции не поддерживается или не поддерживается в широковещательной рассылке
8382	03	Ошибка в длине данных
8383	02	Ошибка в адресе данных или адрес вне действительного диапазона DATA_PTR

STATUS (W#16#....)	Код ответа ведомого устройства	Ошибка протокола Modbus
8384	03	Ошибка в значении данных
8385	03	Значение диагностического кода данных не поддерживается (код функции 08)

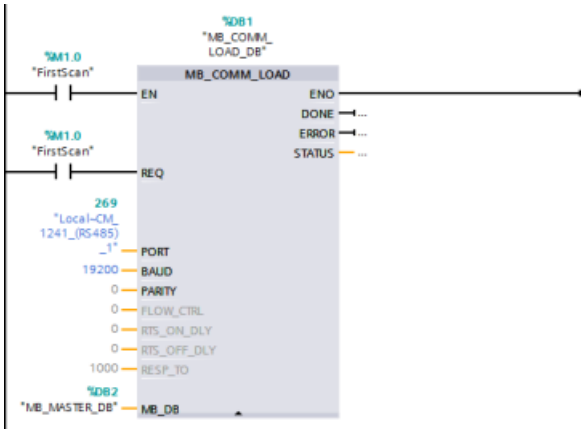
[1] Наряду с перечисленными выше ошибками инструкций MB_SLAVE, базовые PtP коммуникационные инструкции также могут возвращать ошибки.

13.9.4 Пример для Modbus RTU в старых системах

13.9.4.1 Пример: Пример программы для ведущего устройства Modbus RTU в старых системах

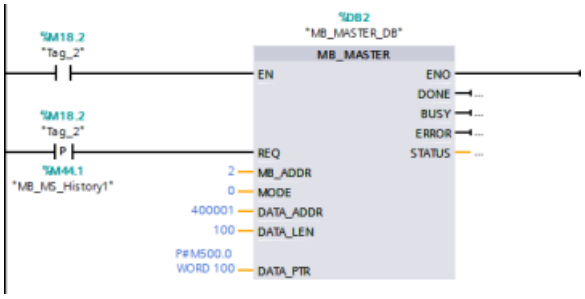
MB_COMM_LOAD инициализируется во время запуска при помощи флага первого цикла. Обработка MB_COMM_LOAD подобным образом должна выполняться только тогда, когда конфигурация последовательного порта не изменяется во время работы.

Сегмент 1: Конфигурировать/инициализировать коммуникационный порт RS485 модуля только один раз во время первого цикла.

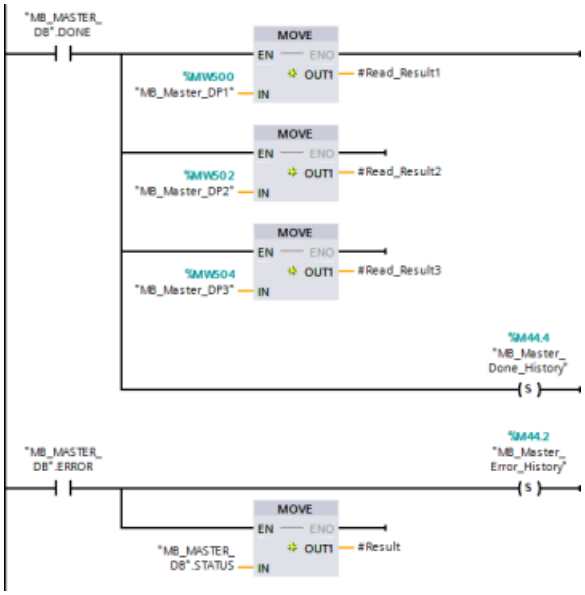


Одна инструкция MB_MASTER используется в OB программного цикла для обмена данными с ведомым устройством. Другие инструкции MB_MASTER могут использоваться в OB программного цикла, чтобы выполнить обмен данными с другими ведомыми устройствами, или один FB MB_MASTER может быть повторно использован, чтобы выполнить коммуникацию с другими ведомыми устройствами.

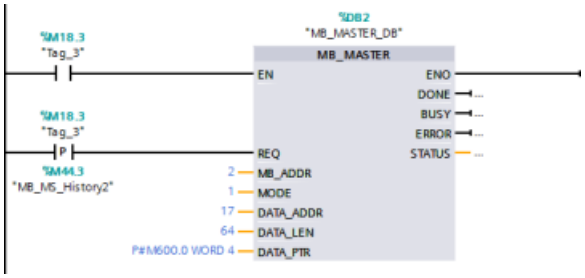
Сегмент 2: Выгрузить 100 слов данных регистра хранения по адресу 400001 на ведомом устройстве № 2 на адрес MW500-MW698.



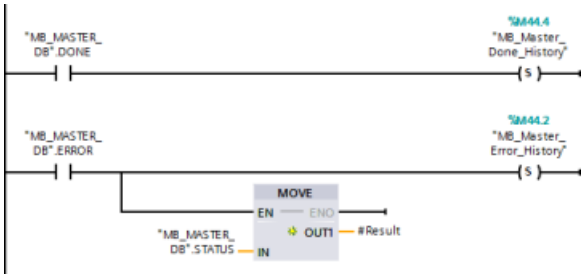
Сегмент 3: Переместить первые 3 слова данных регистра хранения, которые были выгружены на другой адрес, и установить бит истории DONE. Этот сегмент также устанавливает бит истории ERROR и сохраняет слово STATUS по другому адресу в случае ошибки.



Сегмент 4: Записать 64 бита данных из MW600-MW607 по адресам выходных битов с 00017 по 00081 на ведомом устройстве № 2.



Сегмент 5: Установить бит истории DONE после завершения операции записи. При возникновении ошибки программа устанавливает бит истории ERROR и сохраняет код STATUS.

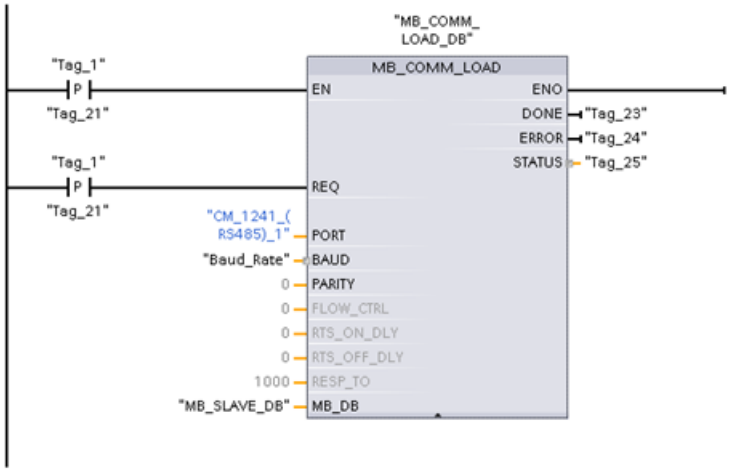


13.9.4.2 Пример: Пример программы для ведомого устройства Modbus RTU в старых системах

Инструкция MB_COMM_LOAD, показанная ниже, инициализируется каждый раз при активации "Tag_1".

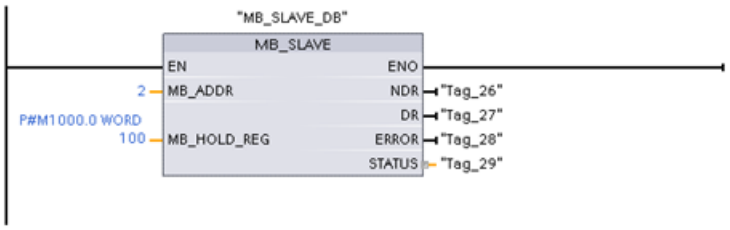
Обработка MB_COMM_LOAD подобным образом должна выполняться только тогда, когда конфигурация последовательного порта изменяется во время работы в результате конфигурирования устройства HMI.

Сегмент 1: Инициализировать параметры модуля RS485 каждый раз, когда они изменяются устройством HMI.



Инструкция MB_SLAVE, показанная ниже, находится в циклическом OB, который выполняется каждые 10 мс. Хотя это и не обеспечивает абсолютно быстрый ответ от ведомого устройства, но гарантирует хорошую производительность при 9600 бодах для коротких сообщений (до 20 байтов в запросе)

Сегмент 2: В каждом цикле проверять наличие запросов от ведущего устройства Modbus. Для регистра хранения Modbus сконфигурировано 100 слов с началом в MW1000.



13.10 Удаленная передача данных в промышленности (Industrial Remote Communication, IRC)

13.10.1 Обзор коммуникационных процессоров для телеуправления

Удаленная передача данных в промышленности предлагает безопасный и рентабельный доступ к удаленным машинам, установкам и приложениям различных конфигураций. Удаленная передача данных в промышленности включает в себя следующие типы коммуникации с использованием CP модулей:

- TeleControl: Телеуправление - это подключение технологических станций (удаленных терминальных блоков, Remote Terminal Unit/ RTU), географически удаленных на значительные расстояния друг от друга, к одной или нескольким центральным системам управления процессом с целью оперативного управления и мониторинга. Различные компоненты передачи в линейке продуктов для удаленных сетей обеспечивают удаленный обмен данными через общедоступные и частные сети. Специальные протоколы телеуправления выполняют событийно-ориентированный или циклический обмен данными процесса для его эффективной организации.
- TeleService: Дистанционное обслуживание (телесервис) включает в себя обмен данными с удаленными техническими системами (машинами, установками, компьютерами и т. д.) с целью обнаружения ошибок, диагностики, технического обслуживания, ремонта или оптимизации.
- Дополнительные приложения для удаленной коммуникации, например, идентификации, приложения для "умных сетей" и мониторинга технического состояния.

Коммуникационные процессоры для телеуправления S7-1200

Для приложений TeleControl предлагаются следующие коммуникационные процессоры, многие из которых также обеспечивают доступ к S7-1200 веб-серверу (Страница 1033):

- **CP 1243-1:**
  - Заказной номер: 6GK7 243-1BX30-0XE0
  - Коммуникационный процессор для подключения SIMATIC S7-1200 через общедоступную инфраструктуру (например, DSL) к центру управления с TeleControl Server Basic (TCSB версии V3).
  - С помощью технологии VPN и брандмауэра, CP обеспечивает защищенный доступ к S7-1200.
  - CP может использоваться в качестве дополнительного Ethernet-интерфейса CPU для S7 коммуникаций.
  - Обмен между CP и CPU осуществляется через конфигурируемые точки данных, которые имеют доступ к переменным (тегам) PLC.
- **CP 1243-1 DNP3:**
  - Заказной номер: 6GK7 243-1JX30-0XE0
  - Коммуникационный процессор для подключения SIMATIC S7-1200 к центрам управления по протоколу DNP3.
  - Обмен между CP и CPU осуществляется через конфигурируемые точки данных, которые имеют доступ к переменным (тегам) PLC.
- **CP 1243-1 IEC:**
  - Заказной номер: 6GK7 243-1PX30-0XE0
  - Коммуникационный процессор для подключения SIMATIC S7-1200 к центрам управления по протоколу IEC 60870-5.
  - Обмен между CP и CPU осуществляется через конфигурируемые точки данных, которые имеют доступ к переменным (тегам) PLC.
- **CP 1243-1 PCC:**
  - Заказной номер: 6GK7 243-1HX30-0XE0
  - Коммуникационный процессор для подключения SIMATIC S7-1200 к центрам управления с помощью Plant Cloud Communication (обмен данными с облаком центра управления, PCC).
  - Обмен между CP и CPU осуществляется через конфигурируемые точки данных, которые имеют доступ к переменным (тегам) PLC.
- **CP 1242-7:**
  - Заказной номер: 6GK7 242-7KX31-0XE0
  - Коммуникационный процессор для подключения SIMATIC S7-1200 через сети сотовой связи (GPRS) и общедоступную инфраструктуру (например, DSL) к центру управления с TeleControl Server Basic.

- **CP 1242-7 GPRS V2:**
  - Заказной номер: 6GK7 242-7KX31-0XE0
  - Коммуникационный процессор для подключения SIMATIC S7-1200 через сети сотовой связи (GPRS) и общедоступную инфраструктуру (например, DSL) к центру управления с TeleControl Server Basic (TCSB версии V3).
  - С помощью технологии VPN и брандмауэра, CP обеспечивает защищенный доступ к S7-1200.
  - CP может использоваться в качестве дополнительного Ethernet-интерфейса CPU для S7 коммуникаций.
  - Обмен между CP и CPU осуществляется через конфигурируемые точки данных, которые имеют доступ к переменным (тегам) PLC.
- **CP 1243-7 LTE-xx:**
  - Коммуникационный процессор для подключения SIMATIC S7-1200 через сети сотовой связи (GPRS) и общедоступную инфраструктуру (например, DSL) к центру управления с TeleControl Server Basic (TCSB версии V3).
  - Поддержка следующих спецификаций сотовой связи: GSM/GPRS, UMTS (G3), LTE
  - Для использования в странах с различными спецификациями сотовой связи, CP доступен в двух вариантах:
    - CP 1243-7 LTE-US:
      - Североамериканский стандарт
      Заказной номер: 6GK7 243-7SX30-0XE0
    - CP 1243-7 LTE-EU:
      - Западноевропейский стандарт
      Заказной номер: 6GK7 243-7KX30-0XE0
  - С помощью технологии VPN и брандмауэра, CP обеспечивает защищенный доступ к S7-1200.
  - CP может использоваться в качестве дополнительного Ethernet-интерфейса CPU для S7 коммуникаций.
  - Обмен между CP и CPU осуществляется через конфигурируемые точки данных, которые имеют доступ к переменным (тегам) PLC.
- **CP 1243-8 IRC:**
  - Заказной номер: 6GK7 242-8RX30-0XE0
  - Коммуникационный процессор для подключения SIMATIC S7-1200 к сети ST7, конфигурируемой точке данных и VPN.

Примечание

Для приложений TeleControl с CP, отличными от CP 1243-1, потребуется программное обеспечение TeleControl Server Basic.

Связь по защищённым сетям

Надежный протокол SINAUT ST7 или стандартизированный протокол DNP3 или IEC 60870-5 обеспечивают защиту для Industrial Remote Communication (http://w3app.siemens.com/mcms/infocenter/dokumentencenter/sc/ic/InfocenterLanguagePacks/Netzwerksicherheit/6ZB5530-1AP02-0BA4_BR_Network_Security_en_112015.pdf). Решение TeleControlпредоставляет весь необходимый инструментарий для предотвращения фальсификации и потери данных. Каждый модуль передачи имеет большую память на несколько тысяч кадров данных, что позволяет перекрывать перебои на линии связи. Выделенные VPN решения защищают специальные IP-сети.

Коммуникационный процессор CP 1243-1 обеспечивает защищенное подключение контроллера SIMATIC S7-1200 с сетями Ethernet. Благодаря встроенному межсетевому экрану (Stateful Inspection) и протоколу VPN (IPsec) коммуникационный процессор помогает защитить станции S7-1200 и сети нижнего уровня от неправомерного доступа и защищает передачу данных от несанкционированных изменений и шпионажа с помощью шифрования. Кроме этого, CP может использоваться для интеграции станции S7-1200 в программное обеспечение центра управления TeleControl Server Basic с использованием удаленных сетей на базе IP.

13.10.2 Подключение к сети GSM

WAN коммуникация на базе IP через GPRS

С помощью коммуникационного процессора CP 1242-7, контроллер S7-1200 может быть подключен к сетям GSM. CP 1242-7 обеспечивает WAN-коммуникацию из удаленных станций с центром управления и межстанционные коммуникации.

Межстанционная коммуникация возможна только через сеть GSM. Для связи между удаленной станцией и центральной диспетчерской, в центре управления должен быть компьютер с доступом к Интернету.

CP 1242-7 поддерживает следующие службы для коммуникации через сеть GSM:

- GPRS (General Packet Radio Service)
  Пакетно-ориентированная служба для передачи данных "GPRS" через сеть GSM.
- Служба коротких сообщений (SMS, Short Message Service)
  CP 1242-7 может получать и отправлять SMS-сообщения. Партнер может быть мобильным телефоном или S7-1200.

CP 1242-7 может использоваться в любых отраслях производства по всему миру и поддерживает следующие диапазоны частот:

- 850 МГц
- 900 МГц
- 1.800 МГц
- 1.900 МГц

Необходимые условия

Оборудование, используемое на станциях или в центре управления, зависит от конкретного приложения.

- Для коммуникаций с центральной диспетчерской или через нее, центру управления необходим компьютер с доступом к Интернету.
- Кроме оборудования станции, удаленная станция S7-1200 с CP 1242-7 должна удовлетворять следующим требованиям, чтобы выполнять коммуникации через сеть GSM:
  - Контракт с подходящим поставщиком услуг сети GSM
    Если используется GPRS, контракт должен разрешать использование службы GPRS. При прямой коммуникации между станциями только через GSM-сеть, поставщик услуг GSM-сети должен присвоить постоянный IP-адрес коммуникационным процессорам. В этом случае обмен данными между станциями выполняется не через центр управления.
  - SIM-карта, прописанная в контракте.
    SIM-карта вставляется в CP 1242-7.
  - Локальная доступность GSM-сети в месте расположения станции

13.10.3 Приложения для CP 1242-7

CP 1242-7 может использоваться в следующих приложениях:

Приложения для телеуправления

- Передача сообщений с помощью SMS
  Через CP 1242-7, CPU удаленной станции S7-1200 получает SMS-сообщения от GSM-сети или передает SMS-сообщения на сконфигурированный мобильный телефон или S7-1200.
- Коммуникации с центром управления
  Удаленные станции S7-1200 выполняют обмен данными через GSM-сеть и Интернет с сервером телеуправления на центральной станции. Для передачи данных с использованием GPRS, приложение "TELECONTROL SERVER BASIC" устанавливается на сервере телеуправления на центральной станции. Сервер телеуправления связывается с центральной системой управления более высокого уровня, используя интегрированную функцию OPC-сервера.
- Коммуникации между станциями S7-1200 через GSM-сеть
  Коммуникация между удаленными станциями с CP 1242-7 может осуществляться двумя различными способами:
  - Межстанционная коммуникация через центр управления
    В этой конфигурации постоянное защищенное соединение между станциями S7-1200, которые связываются друг с другом и сервером телеуправления, устанавливается на центральной станции. Коммуникации между станциями осуществляются через сервер телеуправления. CP 1242-7 работает в режиме "TelecontroI".
  - Прямая коммуникация между станциями
    Для прямого обмена данными между станциями без обхода через центральную станцию используются SIM-карты с постоянным IP-адресом, которые позволяют станциям обращаться друг другу напрямую. Возможные коммуникационные службы и функции безопасности (например, VPN) зависят при этом от оператора сети. CP 1242-7 работает в режиме "GPRS direct".

TeleService через GPRS

Соединение TeleService через GSM-сеть и Интернет может быть установлено между инженерной станцией со STEP 7 и удаленной станцией S7-1200 с CP 1242-7. Соединение работает с инженерной станции через сервер телеуправления или шлюз TeleService, которые выступают в качестве посредников, передавая телеграммы и выполняя авторизацию. Эти PC используют функции приложения TELECONTROL SERVER BASIC.

Соединение TeleService может использоваться в следующих целях:

- Загрузка данных конфигурации или программы из проекта STEP 7 на станцию
- Запрос данных диагностики на станции

13.10.4 Другие возможности CP 1242-7

Другие службы и функции CP 1242-7

- Синхронизация времени CP через Интернет.
  Время на CP может быть установлено следующим образом:
  - В режиме "Telecontrol" время передается сервером телеуправления. CP использует его, чтобы установить свое время.
  - В режиме "GPRS direct" CP может запросить время, используя SNTP.

  Для синхронизации времени CPU, можно считать текущее время из CP, используя блок.
- Временная буферизация отправляемых телеграмм в случае проблем со связью
- Повышенная эксплуатационная готовность благодаря возможности подключения к резервному серверу телеуправления
- Регистрация объема данных
  Объемы переданных данных регистрируются и могут анализироваться.

13.10.5 Дополнительная информация

Подробные сведения можно найти в руководствах/справочниках по коммуникационным процессорам, в соответствующей документации и информации о продуктах:

- CP 1242-7 (http://support.automation.siemens.com/WW/view/en/45605894)
- CP 1243-7 LTE (https://support.industry.siemens.com/cs/ww/en/ps/15924)
- CP 1243-1 DNP3 (https://support.industry.siemens.com/cs/ww/en/ps/15938)
- CP 1243-8 IRC (https://support.industry.siemens.com/cs/ww/en/ps/21162)
- CP 1243-1 IEC (https://support.industry.siemens.com/cs/ww/en/ps/15942)
- Обновления прошивки по мере выпуска (https://support.industry.siemens.com/cs/ww/en/view/109482530)

13.10.6 Аксессуары

GSM/GPRS антенна ANT794-4MR

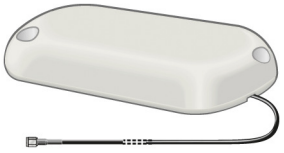
Следующие антенны доступны для использования в сетях GSM/GPRS и могут устанавливаться как в помещениях, так и на открытых площадках:

- Четырехдиапазонная антенна ANT794-4MR (http://support.automation.siemens.com/WW/view/en/23119005)



Условное обозначение	Заказной №	Пояснение
ANT794-4MR	6NH9 860-1AA00	Четырехдиапазонная антенна (900, 1800/1900 МГц, UMTS); с защитой от атмосферных воздействий, для внутренней и наружной установки; в комплекте с антенным кабелем длиной 5 м с соединителем SMA; включая монтажный кронштейн, винты и дюбели

- Плоская антенна ANT794-3M



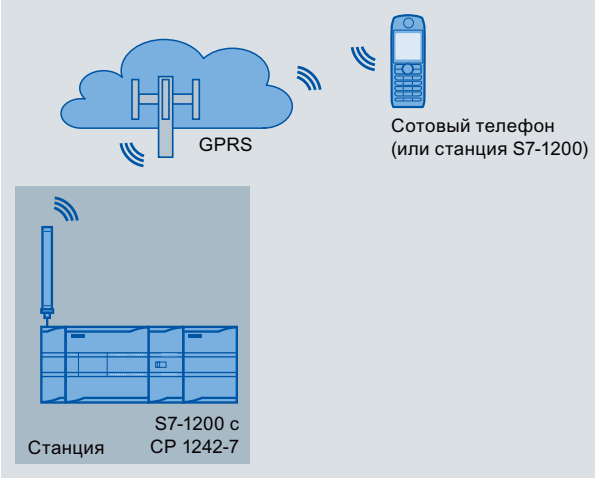
Условное обозначение	Заказной №	Пояснение
ANT794-3M	6NH9 870-1AA00	Плоская антенна (900, 1800/1900 МГц); с защитой от атмосферных воздействий, для внутренней и наружной установки; встроенный антенный кабель длиной 1,2 м со штекером SMA; включая клеящую прокладку, возможно винтовое крепление

Антенны заказываются отдельно.

13.10.7 Примеры конфигурации для телеуправления

Ниже представлено несколько примеров конфигурации для станций с CP 1242-7.

Передача сообщений с помощью SMS



SIMATIC S7-1200 с CP 1242-7 может отправлять SMS-сообщения на мобильный телефон или сконфигурированную станцию S7-1200.

Телеуправление через центр управления

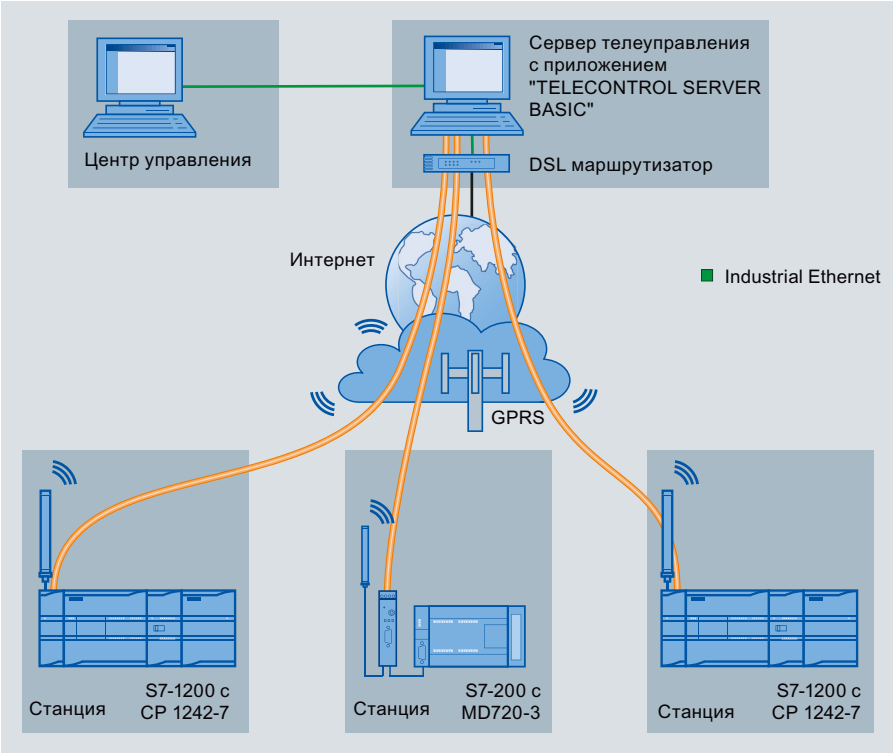


Рисунок 13-29 Коммуникации между станциями S7-1200 и центром управления

В приложениях телеуправления станции SIMATIC S7-1200 с CP 1242-7 выполняют обмен данными с центром управления через GSM-сеть и Интернет. Приложение "TELECONTROL

SERVER BASIC" (TCSB), устанавливается на сервере телеуправления на центральной станции. Это открывает следующие возможности:

- Коммуникационный обмен данными в системах телеуправления между станцией и центром управления
  Вс этом случае данные передаются от удаленных станций через GSM-сеть и Интернет на расположенный на центральной станции сервер телеуправления. Сервер телеуправления контролирует удаленные станции.
- Коммуникации между станцией и диспетчерской с OPC-клиентом
  Как и в первом случае, станции выполняют обмен данным с сервером телеуправления. Сервер телеуправления использует встроенный OPC-сервер для обмена данными с OPC-клиентом диспетчерской.
  OPC-клиент и сервер телеуправления могут быть расположены на одном компьютере, к примеру, если TCSB устанавливается на компьютере центра управления с WinCC.
- Межстанционная коммуникация через центр управления
  Возможен обмен данными с S7 станциями, также оснащенными CP 1242-7.
  Для коммуникации между станциями, сервер телеуправления перенаправляет телеграммы от передающей станции на принимающую станцию.

Прямая коммуникация между станциями

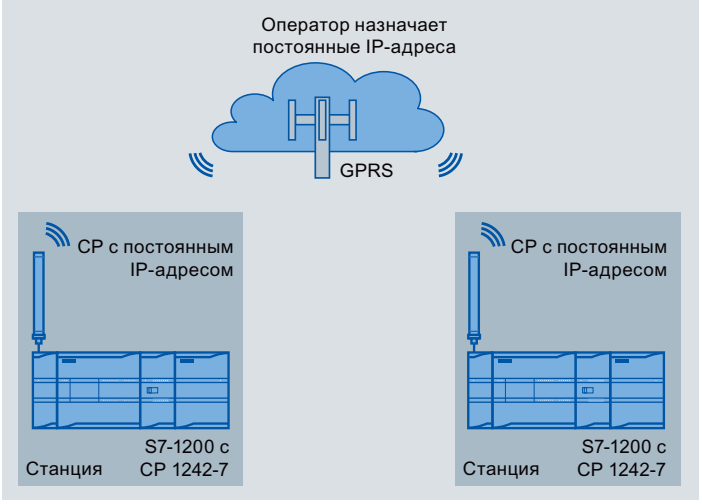


Рисунок 13-30 Прямая коммуникация между двумя станциями S7-1200

В этой конфигурации две станции SIMATIC S7-1200 выполняют прямой обмен данными друг с другом, используя CP 1242-7, через GSM-сеть. У каждого CP 1242-7 есть постоянный IP-адрес. Соответствующая служба поставщика сетевых услуг GSM должна организовать процесс.

TeleService через GPRS

В TeleService через GPRS инженерная станция, на которой установлен STEP 7, обменивается данными по GSM-сети и Интернет со станцией SIMATIC S7-1200 с CP 1242-7.

Так как брандмауэр обычно блокирует внешние запросы на установление соединения, то необходима коммутационная станция между удаленной станцией и инженерной

станцией. Эта коммутационная станция может быть сервером телеуправления или, если в конфигурации нет сервера телеуправления, шлюзом TeleService.

TeleService с сервером телеуправления

Соединение работает через сервер телеуправления.

- Инженерная станция и сервер телеуправления соединены друг с другом через Интранет (LAN) или Интернет.
- Сервер телеуправления и удаленная станция соединены друг с другом через Интернет и GSM-сеть.

Инженерная станция и сервер телеуправления могут быть одним и тем же компьютером; другими словами, STEP 7 и TCSB устанавливаются на одном и том же компьютере.

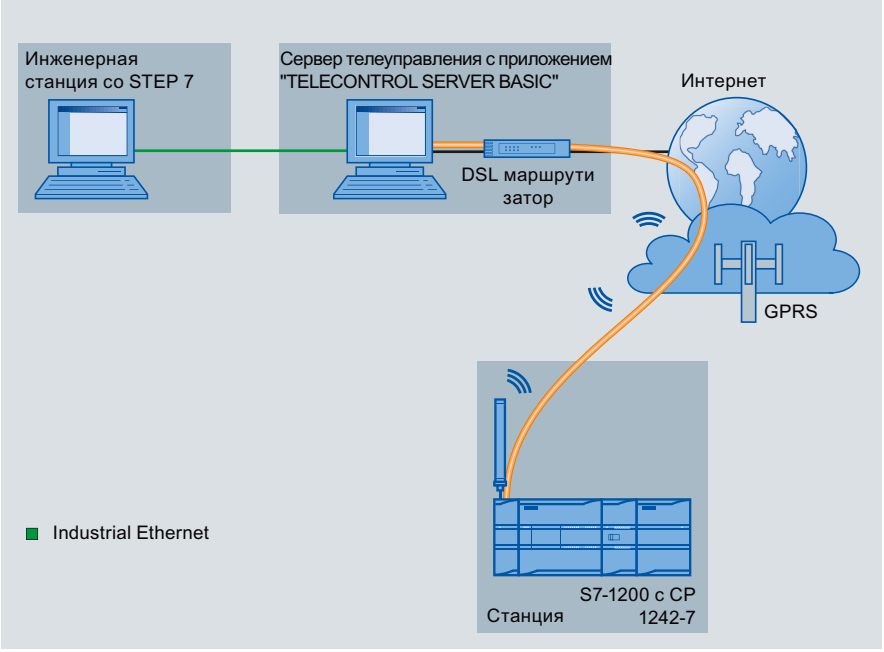


Рисунок 13-31 TeleService через GPRS в конфигурации с сервером телеуправления

TeleService без сервера телеуправления

Соединение работает через шлюз TeleService.

Соединение между инженерной станцией и шлюзом TeleService может быть локальным через LAN или через Интернет.

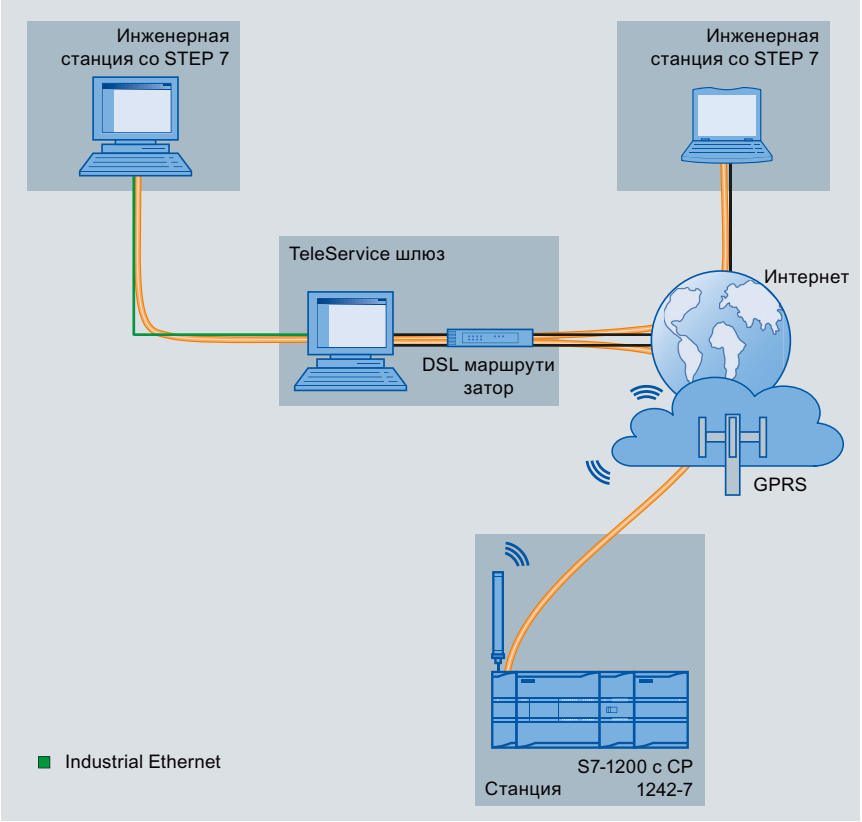
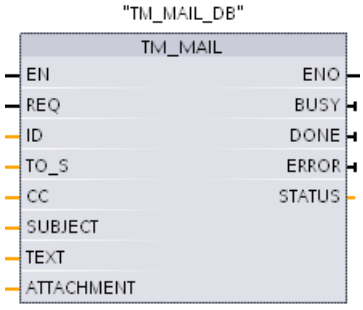


Рисунок 13-32 TeleService через GPRS в конфигурации со шлюзом TeleService

Коммуникации TeleService (SMTP email) 14

14.1 Инструкция TM_Mail (передача электронного сообщения)

Таблица 14-1 Инструкция TM_MAIL

LAD/FBD	SCL	Описание
	**"TM_MAIL_DB"(** **REQ:=_bool_in_,** **ID:=_int_in_,** **TO_S:=_string_in_,** **CC:=_string_in_,** **SUBJECT:=_string_in_,** **TEXT:= _string_in_,** **ATTACHMENT:=_variant_in_,** **BUSY=>_bool_out_,** **DONE=>_bool_out_,** **ERROR=>_bool_out_,** **STATUS=>_word_out_,);**	Инструкция TM_MAIL отправляет электронное письмо, используя SMTP (Simple Mail Transfer Protocol), по TCP/IP через Industrial Ethernet соединение CPU. Там, где Ethernet-подключение к Интернету отсутствует, для соединения с телефонными наземными линиями связи может использоваться дополнительный TS адаптер. TM_MAIL выполняется асинхронно, и задание распределяется по нескольким вызовам TM_MAIL. При вызове TM_MAIL необходимо назначить DB экземпляра. **DB экземпляра не может быть определен в качестве сохраняющегося блока**. Это гарантирует, что DB экземпляра инициализируется при переходе CPU из STOP в RUN и что может быть запущено новое выполнение TM_MAIL.

[1] При добавлении инструкции, STEP 7 автоматически создает блок данных экземпляра.

Операция отправки электронного письма запускается по положительному фронту 0 на 1 на входном параметре REQ. В следующей таблице представлены отношения между BUSY, DONE и ERROR. Можно контролировать прогресс выполнения TM_MAIL и отследить завершение, проанализировав эти параметры в последовательных вызовах.

Выходные параметры, DONE, ERROR, STATUS, и SFC_STATUS действительны только для одного цикла, когда состояние выходного параметра BUSY изменяется из 1 в 0. Программная логика должна сохранить временные значения состояния выхода, чтобы можно было обнаружить изменения состояния в последующих циклах выполнения программы.

Примечание

TM_MAIL отправляет почтовое сообщение по TCP/IP с использованием Ethernet интерфейса CPU. Чтобы отправить почтовое сообщение по CP интерфейсу (с или без SSL), следует использовать инструкцию Инструкция TMAIL_C (передача Email через Ethernet интерфейс CPU) (Страница 891).

Таблица 14-2 Взаимодействие между параметрами Done, Busy и Error

DONE	BUSY	ERROR	Описание
Неприменимо	1	Неприменимо	Задание в процессе обработки.
1	0	0	Задание было успешно выполнено.
0	0	1	Задание было завершено с ошибкой. Причину ошибки можно узнать в параметре STATUS.
0	0	0	Обрабатываемые задания отсутствуют.

Если CPU при выполнении TM_MAIL переходит в режим STOP, то коммуникационное соединение с почтовым сервером прерывается. Коммуникационное соединение с почтовым сервером также теряется, если возникают проблемы в коммуникациях CPU на шине Industrial Ethernet. В этих случаях процесс отправки прерывается, и электронная почта не доходит до получателя.

ВНИМАНИЕ

Изменение программ пользователя

Удаление и замена программных блоков, вызовов TM_MAIL или вызовов DB экземпляра TM_MAIL могут нарушить связь между программными блоками. Без сохранения связанных программных блоков, коммуникационные функции TPC/IP могут перейти в неопределенное состояние и как следствие привести к материальному ущербу. После передачи измененного программного блока необходимо выполнить перезапуск CPU (горячий пуск) или холодный запуск.

Для того, чтобы не прерывать связи между программными блоками, следует изменять части программы пользователя, которые непосредственно влияют на вызовы TM_MAIL, только в следующих случаях:

- CPU находится в состояние STOP
- Отправка почты не выполняется (REQ и BUSY = 0)

Целостность данных

Входной параметр ADDR_MAIL_SERVER считывается при запуске операции. Новое значение не вступает в силу, пока текущая операция не завершена, и инициирована новая операция TM_MAIL.

Напротив, параметры WATCH_DOG_TIME, TO_S, CC, FROM, SUBJECT, TEXT, ATTACHMENT, USERNAME и PASSWORD считываются во время выполнения TM_MAIL и могут быть изменены только после завершения задания (BUSY = 0).

Коммутируемое соединение: конфигурирование IE параметров TS адаптера

Необходимо сконфигурировать IE параметры TS адаптера для исходящих вызовов, чтобы соединиться с коммутируемым сервером Интернет-провайдера. Если установить атрибут для вызова по требованию, то соединение устанавливается только при отправке электронного письма. При использовании аналогового модема требуется больше времени для установления соединения (приблизительно на минуту дольше). Следует учесть это дополнительное время в значении WATCH_DOG_TIME.

Таблица 14-3 Типы данных для параметров

Параметр и тип данных		**Типы данных**	**Описание**
REQ	IN	Bool	Инструкцию запускает положительный фронт (0 на 1).
ID	IN	Int	Идентификатор соединения: См. параметр ID инструкций TCON , TDISCON , TSEND и TRCV. Должно использоваться число, которое не используется для других экземпляров этой инструкции в пользовательской программе.
TO_S	IN	String	Адреса получателей: данные STRING с максимальной длиной в 240 символов.
CC	IN	String	Адреса получателей для CC-копии (опция): данные STRING с максимальной длиной в 240 символов.
SUBJECT	IN	String	Тема сообщения электронной почты: данные STRING с максимальной длиной в 240 символов.
TEXT	IN	String	Текст сообщения электронной почты (опция): данные STRING с максимальной длиной в 240 символов. Если этот параметры содержит пустую строку, то электронное письмо посылается без текстового сообщения.
ATTACHMENT	IN	Variant	Указатель на вложение: Byte, Word или DWord данные с максимальной длиной в 65534 байтов. Если значение не присвоено, письмо будет отправлено без вложения.
DONE	OUT	Bool	• 0 - Задание еще не запущено или еще выполняется. • 1 - Задание выполнено без ошибок.
BUSY	OUT	Bool	• 0 - Нет выполняемых операций • 1 - Операция выполняется
ERROR	OUT	Bool	Бит ERROR в течение одного цикла имеет значение = 1, после того, как последний запрос был завершен с ошибкой. Код ошибки на выходе STATUS действителен только в цикле, где ERROR = 1.
STATUS	OUT	Word	Возвращаемое значение или информация об ошибке инструкции TM_Mail.

Параметр и тип данных		Типы данных	Описание
ADDR_MAIL_SERVER	[1] Static	DWord	IP-адрес почтового сервера: Необходимо указать каждый фрагмент IP-адреса как октет двух 4-разрядных шестнадцатеричных символов. Если фрагмент IP-адреса = десятичное значение 10, который соответствует шестнадцатеричному значению A, то следует ввести "0A" для этого октета. Пример: IP-адрес = 192.168.0.10 ADDR_MAIL_SERVER = DW#16#C0A8000A, где: • 192 = 16#C0, • 168 =16#A8 • 0 = 16#00 • 10 = 16#0A
WATCH_DOG_TIME	[1] Static	Time	Максимально допустимое время для TM_MAIL, чтобы завершить весь SMTP процесс, от инициирования соединения с SMTP до конца SMTP передачи. Если это время превышено, то исполнение TM_MAIL завершается с ошибкой. Фактическая задержка до завершения TM_MAIL и возвращения ошибки может превысить WATCH_DOG_TIME, из-за дополнительного времени, требуемого для разъединения. Сначала следует назначить время в 2 минуты. Это время может быть намного меньшим для телефонного ISDN соединения.
USERNAME	[1] Static	String	Имя пользователя почтовой учетной записи: данные STRING с максимальной длиной в 180 символов.
PASSWORD	[1] Static	String	Пароль почтового сервера: данные STRING с максимальной длиной в 180 символов.
FROM	[1] Static	String	Адрес отправителя: STRING с максимальной длиной в 240 символов.
SFC_STATUS	[1] Static	Word	Код условия для выполнения вызванных коммуникационных блоков

[1] Значения параметров изменяются не при каждом вызове инструкции TM_Mail. Значения назначаются в блоке данных экземпляра TM_MAIL, и к ним происходит обращение только один раз при первом вызове TM_MAIL.

SMTP аутентификация

TM_MAIL поддерживает метод аутентификации SMTP AUTH LOGIN. Для получения информации о данном методе аутентификации, см. инструкцию к почтовому серверу или веб-сайт Интернет-провайдера.

Метод аутентификации AUTH LOGIN использует параметры USERNAME и PASSWORD инструкции TM_MAIL, чтобы соединиться с почтовым сервером. Имя пользователя и пароль должны быть заранее настроены в почтовом ящике на почтовом сервере.

Если никакое значение не присвоено параметру USERNAME, то метод аутентификации AUTH LOGIN не используется, и электронное письмо посылается без аутентификации.

TO_S:, CC: и FROM: параметры

Параметры TO_S:, CC: и FROM: являются строками, как показано в следующих примерах:

TO: <wenna@mydomain.com>, <ruby@mydomain.com>,

CC: <admin@mydomain.com>, <judy@mydomain.com>,

FROM: <admin@mydomain.com>

При вводе этих строк должны соблюдаться следующие правила:

- Должны быть введены символы "TO:", "CC:" и "FROM:", включая двоеточие.
- Перед каждым адресом должны быть введены пробел и открывающая угловая скобка "<". Например, должен быть пробел между "TO:" и <адресом электронной почты>.
- Закрывающая угловая скобка ">" должна быть введена после каждого адреса.
- После каждого адреса в TO_S: и CC: должна вводиться запятая ",". Например, запятая после единственного адреса электронной почты требуется в "TO: <email address>,".
- Только один адрес электронной почты может использоваться для записи FROM:, без запятой в конце.

Из-за режима выполнения и использования памяти, проверка синтаксиса данных TO_S:, CC: и FROM: инструкции TM_Mail не выполняется. Если правила формата, приведенные выше, не соблюдаются точно, то транзакция через SMTP почтовый сервер отклоняется.

Параметры STATUS иSFC_STATUS

Коды условия выполнения, возвращаемые TM_MAIL, могут быть классифицированы следующим образом:

- W#16#0000: Выполнение TM_MAIL было успешно завершено.
- W#16#7xxx: Состояние выполнения TM_MAIL.
- W#16#8xxx: Ошибка во внутреннем вызове коммуникационного устройства или почтового сервера

В таблице ниже предоставлены коды условий для выполнения TM_MAIL, за исключением кодов условий, вызываемых системой коммуникационных модулей.

Примечание

Требования к почтовому серверу

TM_MAIL может соединяться с почтовым сервером, используя SMTP только через порт 25. Назначенный номер порта не может быть изменен.

Большинство IT отделов и внешние почтовые серверы сейчас блокируют порт 25, чтобы не допустить превращения зараженного вирусом PC в опасный генератор электронных сообщений.

Можно соединиться с внутренним почтовым сервером через SMTP и позволить внутреннему серверу управлять текущими средствами обеспечения безопасности, которые необходимы для передачи электронной почты через Интернет на внешний почтовый сервер.

Пример: Конфигурация внутреннего почтового сервера

При использовании Microsoft Exchange в качестве внутреннего почтового сервера, можно сконфигурировать сервер таким образом, чтобы предоставить SMTP доступ с IP-адреса, назначенного целевой системе S7-1200. Сконфигурировать консоль управления Exchange: Server configuration > Hub transport > Receive connectors > IP relay. На вкладке Network есть поле с именем "Receive mail from remote servers that have these IP addresses". Здесь следует ввести IP-адрес устройства PLC, которое выполняет инструкцию TM_MAIL. Аутентификация для этого типа соединения с внутренним сервером Microsoft Exchange не требуется.

Конфигурация почтового сервера

TM_MAIL может использовать только почтовый сервер, который разрешает коммуникацию через порт 25, SMTP и как опция аутентификацию AUTH LOGIN.

Настроить совместимую учетную запись на почтовом сервере с удаленным SMTP входом в систему. Затем отредактировать DB экземпляра для TM_MAIL и ввести строки TM_MAILUSERNAME и PASSWORD , которые используются для аутентификации соединения с почтовым ящиком.

Таблица 14-4 Коды условий

STATUS (W#16#...):	**SFC_STATUS (W#16#...):**	**Описание**
0000	-	Выполнение TM_MAIL было завершено без ошибок Этот STATUS код 0 не гарантирует, что электронное письмо было фактически послано (см. пункт 1 в примечании после таблицы).
7001	-	TM_MAIL активна (BUSY = 1).
7002	7002	TM_MAIL активна (BUSY = 1).
8xxx	xxxx	Обработка TM_MAIL завершена с ошибкой во внутренних вызовах коммуникационной инструкции. Подробная информация о параметре SFC_STATUS находится в описаниях параметра STATUS базовых открытых пользовательских коммуникационных инструкций PROFINET.
8010	xxxx	Не удалось установить соединение: Подробная информация о параметре SFC_STATUS находится в описаниях параметра STATUS инструкции TCON.
8011	xxxx	Ошибка при отправке данных: Подробная информация о параметре SFC_STATUS находится в описаниях параметра STATUS инструкции TSEND.
8012	xxxx	Ошибка при получении данных: Подробная информация о параметре SFC_STATUS находится в описаниях параметра STATUS инструкции TRCV.
8013	xxxx	Не удалось установить соединение: Подробная информация для анализа параметра SFC_STATUS находится в описаниях параметра STATUS инструкций TCON и TDISCON.
8014	-	Не удалось установить соединение: Возможно был введен неправильный IP-адрес почтового сервера (ADDR_MAIL_SERVER), или указано слишком короткое время (WATCH_DOG_TIME) для соединения. Также возможно, что CPU не подключен к сети или имеет место ошибка в конфигурации CPU.
8015	-	Недопустимый указатель для параметра ATTACHMENT: Использовать указатель Variant с присвоением длины и типа данных. Пример: "P#DB.DBX0,0" неправильно, а "P#DB.DBX0,0 Byte 256" правильно.
82xx, 84xx, 85xx	-	Сообщение об ошибке поступает от почтового сервера и соответствует номеру ошибки "8" в протоколе SMTP. См. пункт 2 в примечании после таблицы.

STATUS (W#16#...):	SFC_STATUS (W#16#...):	Описание
8450	-	Операция не выполняется: Почтовый ящик занят, повторить операцию позднее.
8451	-	Операция отменена: Локальна ошибка при обработке. Повторить операцию позднее.
8500	-	Ошибка в синтаксисе команды: Причиной этого также может быть то, что почтовый сервер не поддерживает метод аутентификации LOGIN. Проверить параметры TM_MAIL. Попробовать отправить письмо без аутентификации. Для этого заменить параметр USERNAME пустой строкой.
8501	-	Синтаксическая ошибка: Неправильный параметр или аргумент; возможно, введен неправильный адрес в параметрах TO_S или CC.
8502	-	Команда неизвестна или не реализована: Проверить введенные данные, в частности параметр FROM. Возможно, он заполнен не полностью и не введены символы "@" или ".".
8535	-	Неполная аутентификация SMTP. Возможно, были введены неправильное имя пользователя или пароль.
8550	-	Почтовый сервер недоступен или отсутствуют права доступа. Возможно, были введены неправильное имя пользователя или пароль или почтовый сервер не поддерживает такой тип доступа. Еще одной причиной ошибки может быть ввод неверного имени домена после символа "@" в параметрах TO_S или CC.
8552	-	Операция отменена: Превышен объем выделенной памяти; повторить операцию позднее.
8554	-	Передача не удалась: Повторить операцию позднее.

Примечание

Возможные не выводимые ошибки передачи электронных сообщений

- Неправильное указание адреса получателя не генерирует ошибку STATUS для TM_MAIL. В этом случае нет никакой гарантии, что дополнительные получатели (с корректными адресами электронной почты), получат электронное письмо.
- Дополнительную информацию о кодах ошибок SMTP можно найти в сети Интернет или в документации по ошибкам почтового сервера. Также можно выгрузить последнее сообщение об ошибке почтового сервера. Сообщение об ошибке сохраняется в параметре buffer1 блока данных экземпляра для TM_MAIL.

Онлайн инструменты и диагностика 15

15.1 Светодиодные индикаторы состояния

CPU и модули ввода/вывода используют светодиоды, чтобы предоставить информацию о рабочем состояния модуля или о вводе/выводе.

Светодиодные индикаторы состояния на CPU

CPU предлагает следующие индикаторы состояния:

- STOP/RUN
  - Постоянный желтый свет показывает рабочее состояние STOP
  - Постоянный зеленый свет показывает рабочее состояние RUN
  - Мигающий свет (чередование зеленого и желтого) указывает на то, что CPU находится в режиме STARTUP
- ERROR
  - Мигающий красный светодиод указывает на ошибку, такую как внутренняя ошибка CPU, ошибка карты памяти или ошибка конфигурации (неподходящие модули).
  - Мигающий красный свет в течение трех секунд указывает на более не актуальную ошибку. Например, сброс часов реального времени (RTC) на настройку по умолчанию при отключении питания.
  - Неисправное состояние:
    - Постоянный красный свет указывает на аппаратную неисправность
    - Все светодиоды мигают при обнаружении ошибки прошивкой
- При вставке карты памяти, мигает светодиод MAINT (техобслуживание). Выключить и снова включить CPU. После этого CPU переходит в рабочее состояние STOP. После того, как CPU перешел в режим STOP, выполнить одну из следующих функций, чтобы начать анализ карты памяти:
  - Перевести CPU в режим RUN
  - Выполнить восстановление исходного состояния памяти (MRES)
  - Выключить и снова включить CPU

Состояние светодиодов также можно определить с помощью инструкции LED (Страница 449).

Таблица 15-1 Светодиодные индикаторы состояния для CPU

Описание	**STOP/RUN Желтый/зеленый**	**ERROR Красный**	**MAINT Желтый**
Нет питания	Выкл	Выкл	Выкл
Запуск, самодиагностика или обновление прошивки	Мигание (попеременно желтый/зеленый)	-	Выкл
Рабочее состояние STOP	Вкл (желтый)	-	-
Рабочее состояние RUN	Вкл (зеленый)	-	-
Извлечь карту памяти	Вкл (желтый)	-	Мигает
Ошибка	Вкл (желтый или зеленый)	Мигает	-
Требуется техобслуживание • Форсированный ввод/вывод • Требуется замена батареи (если установлена батарейная плата)	Вкл (желтый или зеленый)	-	Вкл
Аппаратная неисправность	Вкл (желтый)	Вкл	Выкл
Тест светодиодных индикаторов или неисправность прошивки CPU	Мигание (попеременно желтый/зеленый)	Мигает	Мигает
Неизвестная или несовместимая версия конфигурации CPU	Вкл (желтый)	Мигает	Мигает

Примечание

Ошибка "Неизвестная или несовместимая версия конфигурации CPU"

Попытка загрузить программу S7-1200 V3.0 в CPU S7-1200 V4.0 вызывает ошибку CPU, и CPU отображает соответствующее сообщение об ошибке в буфере диагностики. Если причиной была вставка трансферной карты (Страница 140) с неправильной версией программы, то следует извлечь карту, перевести CPU из STOP в RUN, выполнить восстановление исходного состояния памяти (MRES) или выключить и снова включить CPU. Если причиной этого состояния стала загрузка неправильной программы, сбросить CPU на заводские установки (Страница 1393). После устранения состояния ошибки для CPU, можно загрузить правильную программу CPU версии 4.3.

CPU также имеет два светодиода, которые отображают состояние коммуникаций PROFINET. Открыть нижнюю крышку клеммной колодки, чтобы увидеть светодиоды PROFINET.

- Link (зеленый) включается, чтобы показать успешное соединение
- Rx/Tx (желтый) включается, чтобы показать активную передачу

CPU и каждый цифровой сигнальный модуль (SM) имеют светодиод I/O Channel для каждого из цифровых входных и выходных каналов. I/O Channel (зеленый) включается или выключается, указывая на состояние отдельного входа или выхода.

Поведение S7-1200 после критической ошибки

Если прошивка CPU обнаруживает критическую ошибку, она предпринимает попытку перезапуска в режиме ошибки, и, в случае успеха, сигнализирует о режиме ошибки непрерывным миганием светодиодов STOP/RUN, ERROR и MAINT. Пользовательская программа и аппаратная конфигурация не загружаются после перезапуска в режиме ошибки.

Если CPU успешно выполняет перезапуск в режиме ошибки, то CPU выполняет следующие действия:

- Установка выходов CPU и сигнальных плат в 0
- Установка выходов сигнальных модулей в центральной стойке и на распределенной периферии согласно выбору для "Реакция на CPU STOP" в конфигурации устройства для цифровых выходов модуля.

Если перезапуск в режиме ошибки не удается, (например, из-за аппаратного отказа), то светодиоды STOP и ERROR включены, а светодиод MAINT выключен.

⚠ ПРЕДУПРЕЖДЕНИЕ

Нельзя гарантировать безопасную работу в неисправном состоянии

Устройства управления могут выходить из строя в небезопасных рабочих состояниях, вызывая тем самым непредсказуемое поведение управляемого оборудования. Такое неожиданное рабочее состояние системы автоматизации может стать причиной смертельных или тяжких травм и/или материального ущерба.

Поэтому следует позаботиться о функции АВАРИЙНОГО ОСТАНОВА, электромеханических или других резервируемых устройствах обеспечения безопасности, не зависящих от PLC.

Светодиодные индикаторы состояния на сигнальном модуле (SM)

Дополнительно, каждый цифровой SM имеет светодиод DIAG, который показывает состояние модуля:

- Зеленый цвет сообщает о работоспособном состоянии модуля
- Красный цвет, сообщает, что модуль неисправный или неработоспособный

Каждый аналоговый SM имеет светодиод I/O Channel для каждого аналогового входа или выхода.

- Зеленый цвет сообщает, что канал был сконфигурирован и является активным
- Красный цвет, сообщает о состоянии ошибки отдельного аналогового входа или выхода

Дополнительно, каждый аналоговый SM имеет светодиод DIAG, который показывает состояние модуля:

- Зеленый цвет сообщает о работоспособном состоянии модуля
- Красный цвет, сообщает, что модуль неисправный или неработоспособный

SM обнаруживает наличие или отсутствие питания модуля (напряжение со стороны поля, если требуется).

Таблица 15-2 Светодиодные индикаторы состояния для сигнального модуля (SM)

Описание	**DIAG (Красный/зеленый)**	**I/O Channel (Красный/зеленый)**
Напряжение со стороны поля выключено*	Мигающий красный	Мигающий красный
Не сконфигурирован или выполняется обновление	Мигающий зеленый	Выкл
Модуль сконфигурирован без ошибок	Вкл (зеленый)	Вкл (зеленый)
Состояние ошибки	Мигающий красный	-
Ошибка ввода/вывода (с активированной диагностикой)	-	Мигающий красный
Ошибка ввода/вывода (с деактивированной диагностикой)	-	Вкл (зеленый)

* Состояние поддерживается только для аналоговых сигнальных модулей.

Светодиодные индикаторы состояния на сигнальной плате (SB)

Каждая аналоговая SB имеет светодиод I/O Channel для каждого аналогового входа или выхода.

Таблица 15-3 Светодиодные индикаторы состояния для сигнальной платы (SB)

Описание	**I/O Channel (Красный/зеленый)**
Не сконфигурирован или выполняется обновление	Выкл
Плата сконфигурирована без ошибок	Вкл (зеленый)
Ошибка ввода/вывода (с активированной диагностикой)	Мигающий красный
Ошибка ввода/вывода (с деактивированной диагностикой)	Вкл (зеленый)

15.2 Установление онлайн-соединения с CPU

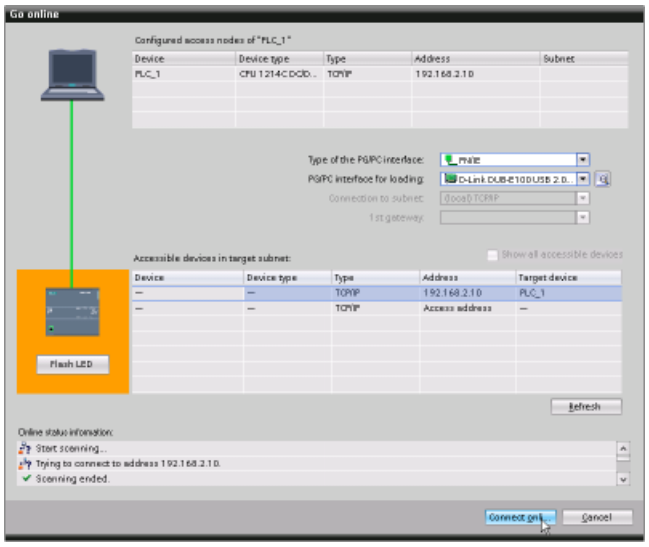
Необходимо установить онлайн соединение между программатором и CPU для загрузки программ и данных проекта, а также для таких действий, как:

- Тестирование программ пользователя
- Отображение и изменение рабочего состояния CPU (Страница 1396)
- Отображение и установка даты и времени CPU (Страница 1393)
- Отображение информации о модуле
- Сравнение и синхронизация (Страница 1399) офлайн и онлайн программных блоков
- Выгрузка и загрузка программных блоков
- Отображение диагностики и буфера диагностики (Страница 1397)

- Использование таблицы текущего контроля (Страница 1404) для тестирования пользовательской программы посредством мониторинга и изменения значений
- Использование таблицы принудительно задаваемых значений для принудительной установки значений в CPU (Страница 1407)

Чтобы установить онлайн-соединение со сконфигурированным CPU, кликнуть по CPU в дереве проекта и нажать кнопку "Установить онлайн-соединение" в проектном представлении:





Если это, первый переход в онлайн с этим CPU, то необходимо выбрать тип PG/PC интерфейса и конкретный PG/PC интерфейс в диалоговом окне "Установить онлайн-соединение" до установки онлайн соединения с CPU на этом интерфейсе.

Программатор должен быть подключен к CPU. Оранжевые рамки указывают на наличие онлайн соединения. Теперь можно использовать инструменты "Онлайн и диагностика" в дереве проекта и карту задач "Онлайн-инструменты".

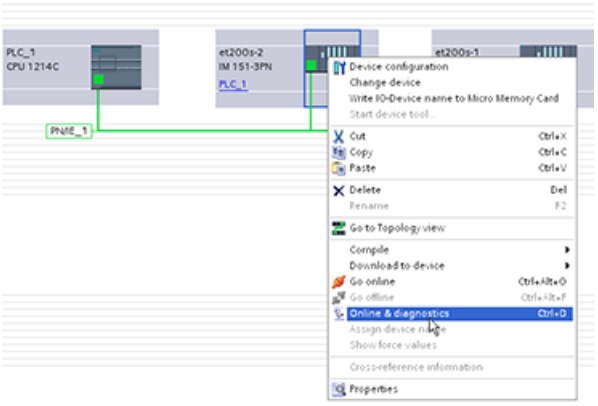
15.3 Присвоение имени устройству PROFINET IO в режиме онлайн

Устройствам в PROFINET-сети должно быть назначено имя, чтобы они могли подключаться к CPU. В редакторе "Устройства и сети" устройствам PROFINET присваиваются имена, если у устройств еще нет имен или если имя устройства должно быть изменено.

Имя PROFINET IO-устройства должно быть присвоено как в проекте STEP 7, так и с помощью инструмента "Онлайн и диагностика" в памяти конфигурации устройства PROFINET IO (например, в памяти конфигурации интерфейсного модуля ET200 S). При

отсутствии имени или несовпадении имен в обоих местах хранения, режим для обмена данными PROFINET IO не будет работать.

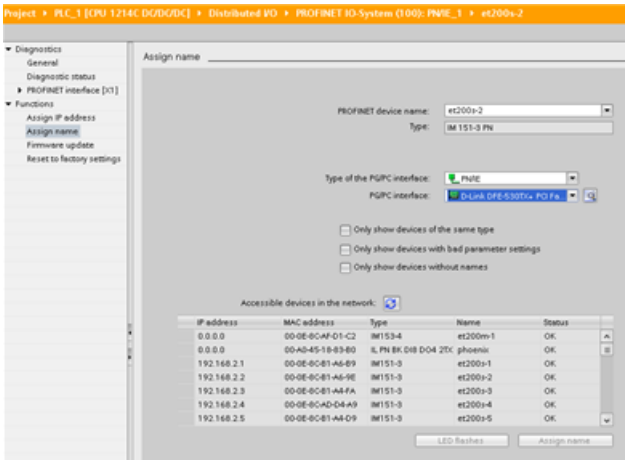
1. Кликнуть правой кнопкой в редакторе "Устройства и сети" по требуемому PROFINET IO устройству и выбрать "Онлайн и диагностика".



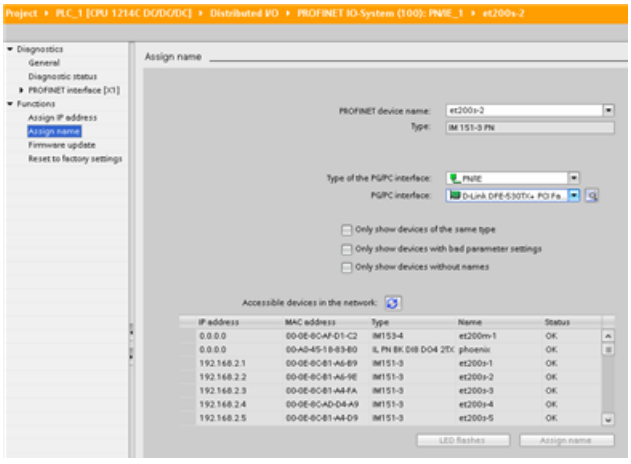
2. Выбрать в диалоговом окне "Онлайн и диагностика" следующие команды меню:

- "Функции":
- "Присвоить имя устройства PROFINET"

Нажать кнопку "Обновить список", чтобы отобразить все PROFINET IO устройства в сети.

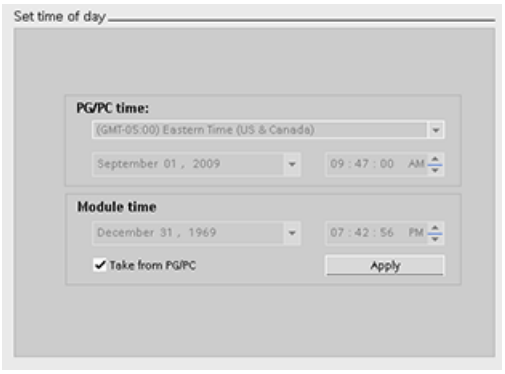


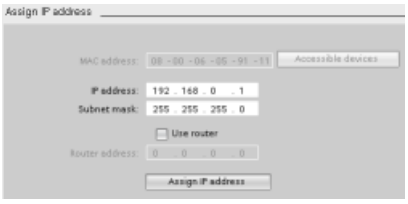
3. В показанном списке кликнуть по требуемому PROFINET IO устройству и после нажать кнопку "Присвоить имя", чтобы записать имя в память конфигурации PROFINET IO устройства.



15.4 Установка IP адреса и времени суток

Можно установить IP-адрес (Страница 817) и время для Онлайн-CPU. После обращения через "Онлайн и диагностика" в деревне проекта Онлайн-CPU, можно показать или изменить IP-адрес. Также можно отобразить или установить параметры времени и даты в Онлайн-CPU.





Примечание

Эта функция доступна только для CPU, у которого есть только MAC-адрес (еще не был назначен IP-адрес), или который был сброшен на заводские настройки.

15.5 Сброс на заводские настройки

Можно выполнить сброс S7-1200 к оригинальным заводским настройкам при следующих условиях:

- У CPU есть онлайн-соединение.
- CPU находится в рабочем состоянии STOP.

Примечание

Если CPU находится в режиме RUN, и активируется процедура сброса, то можно перевести CPU в режим STOP после подтверждения запроса.

Порядок действий

Чтобы выполнить сброс параметров CPU на заводские значения, необходимо выполнить следующие действия:

1. Открыть представление "Онлайн и диагностика" для CPU.
2. Выбрать в папке "Функции" элемент "Возврат к заводским настройкам".
3. Установить флажок "Оставить IP-адрес", если требуется сохранить данный IP-адрес, или установить флажок "Удалить IP-адрес", если IP-адрес необходимо удалить.

4. Нажать кнопку "Сброс".
5. Для подтверждения запроса нажать кнопку "ОК".

Результат

Модуль при необходимости переводится в режим STOP и выполняется сброс параметров на заводские значения: Со стороны CPU выполняются следующие действия:

Со вставленной в CPU картой памяти	Без вставленной в CPU карты памяти
• Удаляется информация из буфера диагностики • Выполняется сброс часов • Восстановление рабочей памяти с карты памяти • Все области операндов устанавливаются на сконфигурированные начальные значения • Все параметры устанавливаются на свои сконфигурированные значения • IP-адрес сохраняется или удаляется согласно сделанному выбору (MAC-адрес является постоянным и никогда не изменяется.)[1] • Набор управляющих данных (Страница 157) удаляется, если он существует	• Удаляется информация из буфера диагностики • Выполняется сброс часов • Удаляется информация из рабочей памяти и внутренней загружаемой памяти • Все области операндов устанавливаются на сконфигурированные начальные значения • Все параметры устанавливаются на свои сконфигурированные значения • IP-адрес сохраняется или удаляется согласно сделанному выбору (MAC-адрес является постоянным и никогда не изменяется.)[1] • Набор управляющих данных удаляется, если он существует

[1] Если был выбран вариант "Оставить IP-адрес", то CPU устанавливает IP-адрес, маску подсети и адрес маршрутизатора (если используется) на параметры из аппаратной конфигурации, если эти значения не были изменены из пользовательской программы или с помощью другого инструмента. В этом случае CPU снова восстанавливает измененные значения.

15.6 Обновление прошивки

Можно обновить прошивку у подключенного CPU из STEP 7 с помощью инструментов онлайн и диагностики одним из следующих способов:

- Обновление через CPU в проекте
- Обновление через доступные устройства в дереве проекта

Обновление прошивки CPU в проекте

Для обновления прошивки выполнить следующие действия:

1. Открыть CPU в дереве проекта, соответствующем подключенному CPU.
2. Открыть просмотр онлайн и диагностики для подключенного CPU.
3. Выбрать в папке "Функции" элемент "Обновление прошивки".

4. В области "Загрузчик прошивки" нажать кнопку "Просмотр", чтобы перейти к файлу обновления прошивки. Он может находиться на жестком диске в месте, куда обновление прошивки S7-1200 (http://support.automation.siemens.com/WW/view/en/34612486/133100) было скачано с веб-сайта онлайн-поддержки для промышленности Siemens (http://support.industry.siemens.com) и сохранено.
5. Выбрать подходящий для используемого модуля файл. В таблице отображаются совместимые модули для выбранного файла.
6. Нажать на кнопку "Выполнить обновление". При необходимости придерживаться указаний в диалоговых окнах, чтобы изменить рабочий режим CPU.

При загрузке обновления прошивки, STEP 7 показывает прогресс этой операции. После завершения загрузки предлагается запустить модуль с новой прошивкой.

Примечание

Если не запустить модуль с новой прошивкой, то предыдущая прошивка остается активной до сброса модуля, например, через выключение и включение питания. Только после сброса модуля новая прошивка начинает работать.

Обновление прошивки через доступные устройства

Для обновления прошивки одного или нескольких доступных устройств выполнить следующие действия:

1. Открыть "Онлайн-доступ" в дереве проекта.
2. Открыть коммуникационный интерфейс, к которому подключен CPU.
3. Двойной клик по "Обновить доступные устройства" и ожидать, пока STEP 7 не отобразит онлайн-устройства.
4. Развернуть CPU, который необходимо обновить, и выполнить двойной клик по "Онлайн и диагностика".
5. Раскрыть в папке "Функции" элемент "Обновление прошивки". Отображаются PLC и локальные модули для PLC. Через область "PLC" или область "Локальные модули" можно продолжить обновление прошивки согласно описанию выше в области "Загрузчик прошивки".

Обновление прошивки может быть выполнено и одним из следующих способов:

- С помощью карты памяти SIMATIC (Страница 147)
- Через стандартную веб-страницу "Информация о модулях" веб-сервера (Страница 1046)
- С помощью SIMATIC Automation Tool (https://support.industry.siemens.com/cs/ww/en/view/98161300)

15.7 Форматирование карты памяти SIMATIC через STEP 7

Можно отформатировать карту памяти в подключенном CPU с помощью инструментов онлайн и диагностики в STEP 7. Выполнить следующие действия:

1. Убедиться, что CPU находится в состояние STOP. Если CPU находится в рабочем состоянии RUN и запускается процесс форматирования, то появляется запрос от STEP 7 на подтверждение перехода CPU в STOP.
2. Вставить карту памяти в подключенный CPU.
3. Открыть для подключенного CPU опцию Онлайн и диагностика через CPU в проекте или через доступные устройства в пункте Онлайн-доступ в дереве проекта.
4. Если CPU не в режиме онлайн, выбрать для подключенного CPU команду "Установить онлайн-соединение".
5. Выбрать в меню "Функции" опцию "Форматировать карту памяти".
6. Кликнуть по "Форматировать".
7. Подтвердить сообщение с "Да".

После STEP 7 форматирует карту памяти и отображает по завершении процесса сообщение в окне информации. CPU после завершения процесса форматирования находится в рабочем состоянии STOP и светодиоды STOP и MAINT мигают. В это время переход в рабочее состояние RUN невозможен. Возможны следующие действия:

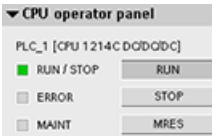
- Извлечение карты памяти и перезапуск CPU: Если внутренняя загружаемая память CPU содержит программу, то CPU запускается с этой программой.
- Перезапуск CPU без извлечения карты памяти: Если внутренняя загружаемая память CPU содержит программу, то CPU копирует эту программу на карту памяти и запускается с этой программой. Если во внутренней загружаемой памяти нет программы, то CPU изменяет тип карты памяти на программную карту (Страница 142) и ожидает загрузку.

Примечание

Форматирование карты памяти не сказывается на содержимом внутренней загружаемой памяти.

Если на момент вставки карты памяти CPU использует внутреннюю загружаемую память и в промежутке времени между вставкой карты памяти и выполнением форматирования не выполнялась перезагрузка CPU, то CPU продолжает сохранять содержимое внутренней загружаемой памяти.

15.8 Панель оператора для Онлайн-CPU



На панели оператора CPU отображается рабочее состояние (STOP или RUN) для Онлайн-CPU. Также отображается наличие ошибок в CPU и принудительное присваивание значений.

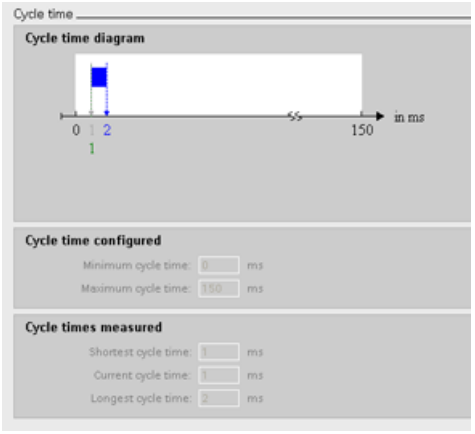
На панели оператора CPU карты задач "Онлайн-инструменты" можно изменять рабочее состояние Онлайн-CPU. Карта задач "Онлайн-инструменты" доступна при нахождении CPU в режиме онлайн.

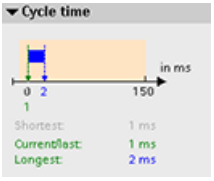
15.9 Мониторинг времени цикла и использования ресурсов памяти

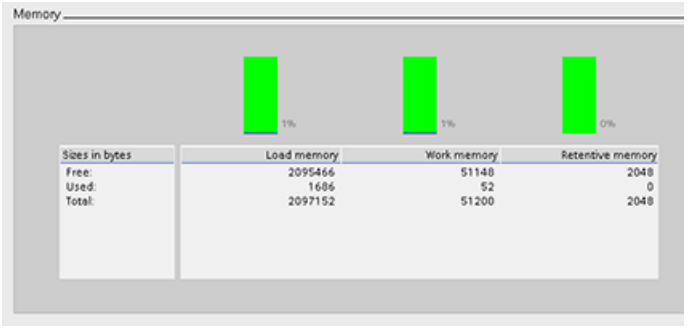
Можно контролировать время цикла и использование ресурсов памяти для Онлайн-CPU.

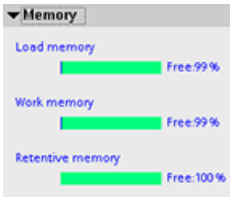
После соединения с Онлайн-CPU открыть карту задач "Онлайн-инструменты", чтобы отобразить следующие измеренные значения:

- Время цикла
- Использование памяти





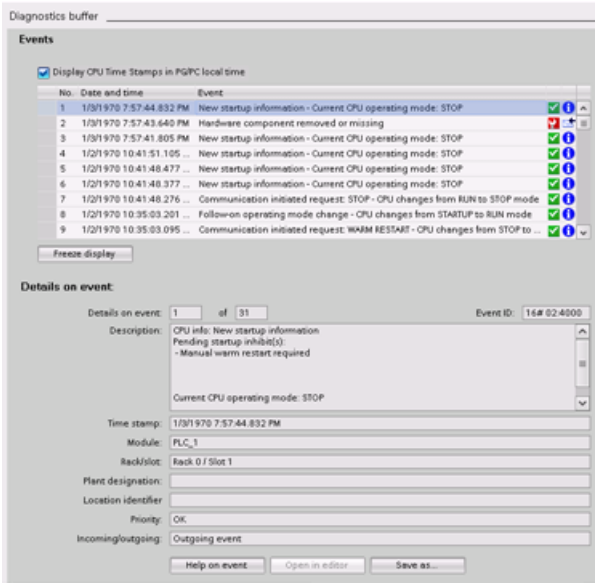




15.10 Отображение диагностических событий в CPU

Используя буфер диагностики, можно просмотреть последние события в CPU. Буфер диагностики доступен в дереве проекта через "Онлайн и диагностика" для Онлайн-CPU. Он содержит следующие элементы:

- Диагностические события
- Изменения в рабочем состоянии CPU (переключение между STOP и RUN)



Первый элемент соответствует последнему событию. Каждый элемент содержит дату и время, когда событие было зарегистрировано, и описание.

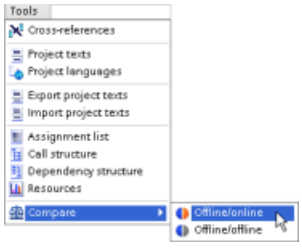
Возможное количество элементов зависит от CPU. Поддерживается до 50 элементов.

Только последние 10 событий в буфере диагностики сохраняются энергонезависимо. При сбросе CPU на заводские настройки, элементы в буфере диагностики удаляются.

Для сбора диагностической информации можно использовать и инструкцию GET_DIAG (Страница 479).

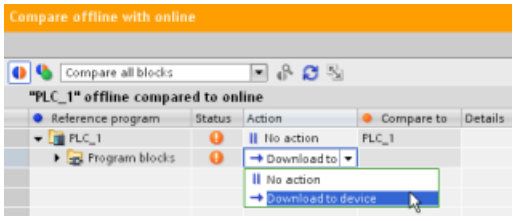
15.11 Сравнение офлайн и онлайн CPU

Можно сравнивать блоки кода в Онлайн-CPU с блоками кода в проекте. Если блоки кода в проекте не соответствуют блокам кода в Онлайн-CPU, то редактор "Сравнение" позволяет синхронизировать проект с Онлайн-CPU. Для этого можно либо загружать блоки кода из проекта в CPU, либо удалять из проекта блоки, отсутствующие в Онлайн-CPU.



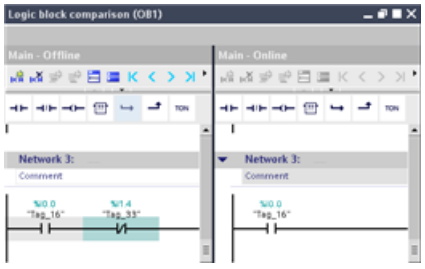
Выбрать CPU в проекте.

Выбрать в редакторе "Сравнение" командуерите команду "Сравнить офлайн/онлайн". (Для вызова команду можно использовать меню "Инструменты" или клик правой кнопкой по CPU в проекте.)



Кликнуть в столбце "Действие" объекта, чтобы выбрать, удалить ли объект, не предпринимать никаких действий или загрузить объект в устройство.

После нажатия на кнопку "Синхронизировать", выполняется загрузка блоков кода.



Кликнуть правой кнопкой по объекту в столбце "Сравнить с" и выбрать "Запустить подробное сравнение", чтобы отобразить блоки кода друг рядом с другом.

Подробное сравнение показывает различия между блоками кода Онлайн-CPU и блоками кода CPU в проекте.

Примечание

Доступ по чтению к защищенному CPU является необходимым условием для сравнения офлайн/онлайн

Для STEP 7 V14 или выше уровня безопасности "Доступ через HMI" недостаточно для сравнения офлайн/онлайн. Для сравнения офлайн/онлайн потребуется "Доступ по чтению" или "Полный доступ".

См. также Защита от несанкционированного доступа для CPU (Страница 203)

15.12 Сравнение топологии онлайн/офлайн

В обзоре топологий в STEP 7 можно сравнить сконфигурированную офлайн-топологию с фактической онлайн-топологией.

Порядок действий

Для определения различий между сконфигурированной и фактической топологией, выполните следующие действия:

1. Вызвать таблицу обзора топологий в представлении топологий.
2. Нажать на панели инструментов в обзоре топологий на кнопку "Сравнение офлайн/онлайн": 

Результат

STEP 7 удаляет столбцы "Удаленная станция", „Удаленный интерфейс“ и „Данные кабеля" в обзорной таблице топологий и вставляет столбы для сравнения для "Состояние" и "Действие". Статус сравнения отображается для каждого устройства или каждого порта в обзоре топологий следующим образом:

Символ	Значение
	Различия в топологиях минимум по одному подчиненному компоненту
	Идентичные топологии
	Информация о топологии доступна только офлайн или устройство выключено
	Информация о топологии доступна только онлайн
	Разница в топологии
	Устройство не поддерживает топологические функции

Столбец действия предлагает для каждого сравненного порта или каждого сравненного устройства следующие опции:

Символ	Значение
	Действия невозможны
	Принять онлайн-соединение

Для повторения сравнения нажать на панели инструментов в обзоре топологий на кнопку .

Дополнительная информация о представлении топологий и сравнении топологий онлайн/офлайн находится в информационной системе STEP 7. В справочнике PROFINET со STEP 7 V13 (https://support.industry.siemens.com/cs/ww/en/view/49948856) также можно найти дополнительную информацию".

15.13 Мониторинг и изменение значений в CPU

STEP 7 предлагает онлайн-функции для мониторинга CPU:

- Можно отобразить или отслеживать текущие значения переменных (тегов). Функция мониторинга не вмешивается в выполнение программы. Она предоставляет информацию о выполнении программы и данных программы в CPU.
- Можно также использовать и другие функции, чтобы управлять последовательностью и данными пользовательской программы:
  - Можно изменять значение для переменных (тегов) в Онлайн-CPU, чтобы увидеть реакцию пользовательской программы.
  - Можно принудительно устанавливать периферийный выход (к примеру, Q0.1:P или "Start":P) на определенное значение.
  - Можно разрешить выходы в режиме STOP.

Примечание

Следует всегда проявлять осторожность при использовании функций управления. Эти функции могут серьезно повлиять на выполнение пользовательской/системной программы.

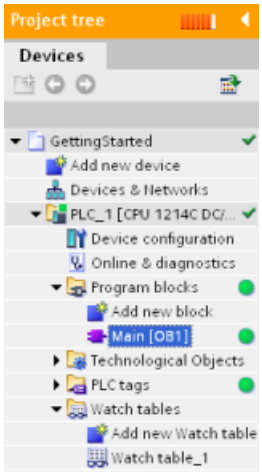
Таблица 15-4 Онлайн-функции редакторов STEP 7

Редактор	Мониторинг	Изменение	Принудительное присваивание значений
Таблица текущего контроля	Да	Да	Нет
Таблица принудительно задаваемых значений	Да	Нет	Да
Редактор текстов программ	Да	Да	Нет
Таблица переменных (тегов)	Да	Нет	Нет
Редактор блоков данных	Да	Нет	Нет

15.13.1 Переход в онлайн для наблюдения за значениями в CPU

Мониторинг переменных (тегов) предполагает наличие онлайн-соединения с CPU. На панели инструментов нажать кнопку "Установить онлайн-соединение".

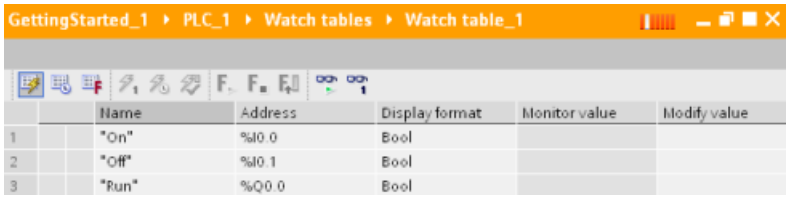




После соединения с CPU, STEP 7 выделяет заголовки рабочих областей оранжевым цветом.

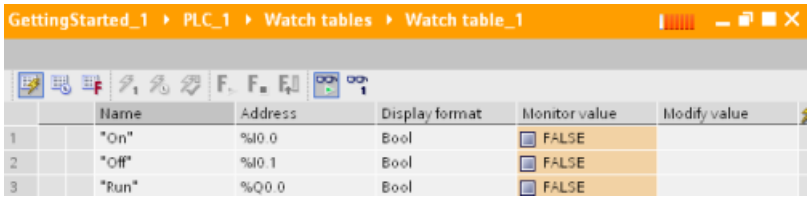
Дерево проекта отображает сравнение офлайн проекта и онлайн CPU. Зеленый кружок означает, что CPU и проект синхронизированы, т.е., что они имеют ту же конфигурацию и пользовательскую программу.

Таблицы переменных (тегов) содержат переменные (теги). Таблицы текущего контроля также могут отображать переменные (теги), как и прямые адреса.





Чтобы наблюдать выполнение пользовательской программы и отобразить значения переменных (тегов), нажать на кнопку "Контролировать все" на панели инструментов.



В поле "Контролируемое значение" для каждой переменной (тега) отображается значение.

15.13.2 Отображение состояния в редакторе текстов программ

Можно наблюдать за состоянием до 50 переменных (тегов) в и редакторе текстов программ LAD и FBD. Использовать панель редакторов, чтобы вывести на экран редактор LAD. Панель редакторов позволяет переключаться между открытыми редакторами без необходимости открывать или закрывать редакторы.

Нажать в редакторе текстов программы на панели инструментов на кнопку "Вкл/выкл мониторинг", чтобы отобразить состояние для программы пользователя.





В сегменте в редакторе текстов программы поток сигналов отображается зеленой линией.

Можно кликнуть правой кнопкой по инструкции или параметру, чтобы изменить значение для инструкции.

15.13.3 Регистрация мгновенных онлайн-значений DB для восстановления значений

Можно сделать мгновенный снимок фактических значений для переменных (тегов) блока данных в Онлайн-CPU для использования этих значений в будущем.

Действуют следующие ограничения:

- Требуется онлайн-соединение с CPU.
- DB должен быть открыт в STEP 7.

Создание мгновенного снимка

Для создания мгновенного снимка выполнить следующие действия:

1. Нажать в редакторе DB на кнопку "Контролировать все переменные (теги)". Столбец "Контролируемое значение" показывает фактические значения данных.
2. Нажать на кнопку , чтобы сделать мгновенный снимок фактических значений и отобразить его в столбце "Мгновенный снимок".

Этот мгновенный снимок можно использовать в будущем для обновления фактических значений или для замены стартовых значений.

Копирование значений мгновенного снимка в CPU

Для копирования значений мгновенного снимка в фактические значения переменных (тегов) блоков в CPU, нажать следующую кнопку:

Онлайн-CPU загружает значения мгновенного снимка в фактические значения. Столбец "Контролируемое значение" показывает фактические значения в CPU. Позднее цикл может изменить значения в CPU, но на момент копирования CPU загружает значения мгновенного снимка в рамках согласованного процесса.

Примечание

Если моментный снимок содержит информацию о состоянии, значения времени или вычисленную информацию, то CPU восстанавливает эти значения такими, какими они были на момент создания снимка.

Копирование значений мгновенного снимка в стартовые значения

Для копирования значений мгновенного снимка в стартовые значения переменных (тегов) блоков данных в CPU, нажать следующую кнопку: 

После компиляции DB и загрузки его в CPU, DB использует новые стартовые значения при переходе CPU в рабочее состояние RUN.

Копирование отдельных значений мгновенного снимка или контролируемых значений в стартовые значения

В редакторе блока данных можно копировать и отдельные значения с последующей их вставкой в стартовые зачения. Кликнуть правой кнопкой мыши по значению в любом столбце и выбрать "Копировать", чтобы поместить значение в буфер. После можно кликнуть правой кнопкой по любому стартовому значению и выбрать "Вставить", чтобы заменить его значение на значение из буфера.

После компиляции DB и загрузки его в CPU, DB использует новые стартовые значения при переходе CPU в рабочее состояние RUN.

15.13.4 Мониторинг и изменение значений в CPU через таблицу текущего контроля

Таблица текущего контроля позволяет выполнять функции мониторинга и управления в точках данных, при обработке CPU программы. Эти точки данных могут быть образом процесса (I или Q), M, DB или физическими входами (I_:P), в зависимости от функции мониторинга или управления. Нельзя строго контролировать физические выходы (Q_:P), потому что функция мониторинга может отобразить только последнее значение, записанное из Q памяти, и не читает фактическое значение из физических выходов.

Функция мониторинга не вмешивается в выполнение программы. Она предоставляет информацию о выполнении программы и данных программы в CPU.

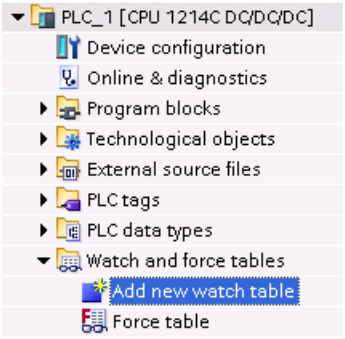
Функции управления позволяют пользователю управлять ходом и данными программы. Следует всегда проявлять осторожность при использовании функций управления. Эти функции могут серьезно повлиять на выполнение пользовательской/системной программы. В случае этих трех функций управления речь идет об изменении, принудительном присваивании значений и разрешении выходов в STOP.

Таблица текущего контроля позволяет выполнять следующие онлайн-функции:

- Мониторинг состояния переменных (тегов)
- Редактирование значений отдельных переменных (тегов)

Можно выбирать, когда необходимо выполнять мониторинг или редактирование переменной (тега):

- Начало цикла: Значение считывается или записывается в начале цикла
- Конец цикла: Значение считывается или записывается в конце цикла
- Переключить на Стоп



Создание таблицы текущего контроля:

1. Открытие двойным кликом по "Добавить новую таблицу текущего контроля" новую таблицу текущего контроля.
2. Ввести имя переменной (тега), чтобы добавить переменную (тег) в таблицу текущего контроля.

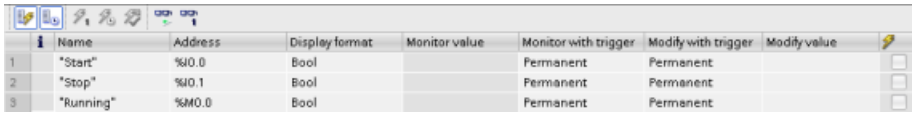
Для мониторинга переменных (тегов) доступны следующие возможности:

- Контролировать все Эта команда запускает мониторинг видимых переменных (тегов) в активной таблице текущего контроля.
- Контролировать сейчас: Эта команда запускает мониторинг видимых переменных (тегов) в активной таблице текущего контроля. Таблица текущего контроля начинает мониторинг переменных (тегов) сразу же и однократно.

Для редактирования переменных (тегов) доступны следующие возможности:

- "Изменить на 0" устанавливает значение выбранного адреса на "0".
- "Изменить на 1" устанавливает значение выбранного адреса на "1".
- "Изменить сейчас" изменяет значение выбранного адреса сразу же и только на один цикл.
- "Изменить с триггером" изменяет значения выбранных адресов.
  Эта функция не создает подтверждения для изменения выбранных адресов. Если требуется подтверждение, то следует использовать функцию "Изменить сейчас".
- Функция "Включить периферийные выходы" отключает команду на блокировку выходов и доступна только для CPU в рабочем состоянии STOP.

Но такой мониторинг переменных (тегов) предполагает наличие онлайн-соединения с CPU.



	Name	Address	Display format	Monitor value	Monitor with trigger	Modify with trigger	Modify value	
1	"Start"	%I0.0	Bool		Permanent	Permanent		
2	"Stop"	%I0.1	Bool		Permanent	Permanent		
3	"Running"	%M0.0	Bool		Permanent	Permanent		

С помощью кнопок над таблицей текущего контроля можно выбирать различные функции.

Ввести имя переменной (тега) для мониторинга и выбрать формат отображения из выпадающего списка. При наличии онлайн соединения с CPU нажать кнопку "Контролировать", чтобы отобразить фактическое значение точки данных в поле "Контролируемое значение".

15.13.4.1 Использование триггера при мониторинге или изменении переменных (тегов) PLC

Запуск по триггеру определяет, в какой точке цикла будет контролироваться или изменяться выбранный адрес.

Таблица 15-5 Типы триггеров

Триггер	Описание
Постоянно	Данные регистрируются непрерывно
В начале цикла	Постоянно: Данные регистрируются в постоянном режиме в начале цикла, после того, как CPU выполнил чтение входов
	Однократно: Данные регистрируются в начале цикла, после того, как CPU выполнил чтение входов
В конце цикла	Постоянно: Данные регистрируются в постоянном режиме в конце цикла, до того, как CPU выполнит запись на выходы
	Однократно: Данные регистрируются в конце цикла, до того, как CPU выполнит запись на выходы
При переходе в STOP	Постоянно: Данные регистрируются постоянно при переходе CPU в STOP
	Однократно: Данные регистрируются однократно после перехода CPU в STOP

Для изменения PLC переменной (тега) с заданным триггером, выбрать начало или конец цикла.

- Изменение выхода: Наилучшим моментом для изменения выхода является конец цикла, непосредственно перед записью CPU на выходы.
  Наблюдать за значением выходов в начале цикла, чтобы определить какое значение записывается на физические выходы. Кроме того, наблюдать за выходами, прежде чем CPU запишет значения на физические выходы, чтобы проверить логику программы и сравнить с поведением фактического ввода-вывода.
- Изменение входа: Наилучшим моментом для изменения входа является начало цикла, сразу после того, как CPU прочтет входы и до того, как пользовательская программа будет использовать входные значения.
  Если есть подозрение, что значения изменяются во время цикла, то имеет смысл наблюдать значение входов в конце цикла, чтобы гарантировать, что значения не изменились за цикл. Если есть различие в значениях, то пользовательская программа, возможно, ошибочно записывает не на тот вход.

Чтобы определить возможную причину перехода CPU в STOP, следует использовать триггер "Переход в STOP", чтобы зарегистрировать последние значения процесса.

15.13.4.2 Разрешение выходов в режиме STOP

Таблица текущего контроля позволяет выполнять запись на выходы, когда CPU находится в режиме STOP. Эта функциональность позволяет проверить проводные соединения выходов и убедиться, что провод, соединенный с выходным контактом, передает сигнал 1 или 0 на клемму устройства, с которым он соединен.

⚠ ПРЕДУПРЕЖДЕНИЕ

Риски, связанные с записью на физические выходы в режиме STOP

Даже при условии, что CPU находится в режиме STOP, разрешение физического выхода может активировать точку процесса, к которой этот выход подключен, вызвав непредсказуемую работу оборудования. Это может привести к смерти или тяжелым телесным повреждениям.

Прежде, чем выполнить запись на выход в таблице текущего контроля, следует убедиться, что изменение значения физического выхода не приведет к непредсказуемой работе оборудования. Всегда необходимо соблюдать технику безопасности для технологического оборудования.

Можно изменить состояние выходов в режиме STOP, когда выходы активированы. Если выходы деактивированы, нельзя изменять выходы в режиме STOP. Чтобы разрешить изменение выходов в режиме STOP из таблицы текущего контроля, выполнить следующие шаги:

1. В меню "Онлайн" выбрать команду "Расширенный режим".
2. Выбрать в меню "Онлайн" команду "Изменить" и после опцию "Включить периферийные выходы" или кликнуть в контекстном меню правой кнопкой мыши по соответствующей строке в таблице текущего контроля.

Если была сконфигурирована распределенная периферия, то нельзя активировать выходы в рабочем режиме STOP. При попытке появляется сообщение об ошибке.

При переводе CPU в режим RUN опция "Включить периферийные выходы" отключается.

Если для входов или выходов принудительно назначены значения, то CPU в рабочем состоянии STOP не может активировать выходы. Сначала необходимо завершить функцию для принудительного назначения значений.

15.13.5 Принудительная установка значений в CPU

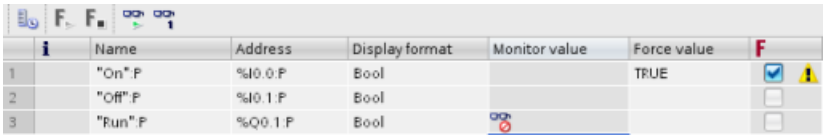
15.13.5.1 Работа с таблицей принудительно задаваемых значений

Таблица принудительно задаваемых значений предлагает функцию "принудительной установки", которая устанавливает значение для входа или выхода на значение для адреса периферийного входа или выхода. Принудительное присваивание значений выполняется в образе процесса входов до выполнения пользовательской программы и в образе процесса выходов, прежде чем выходы будут записаны в модули.

Примечание

Принудительно задаваемые значения сохраняются в CPU, а не в таблице принудительно задаваемых значений.

Нельзя принудительно присвоить значение входу (адрес "I") или выходу (адрес "Q"). Но можно принудительно присвоить значение периферийному входу или выходу. Таблица принудительно задаваемых значений автоматический прикрепляет ":P" к адресу (пример: "On":P или "Run":P).



	i	Name	Address	Display format	Monitor value	Force value	F
1		"On":P	%I0.0:P	Bool		TRUE	☑ ⚠
2		"Off":P	%I0.1:P	Bool			☐
3		"Run":P	%Q0.1:P	Bool			☐

Ввести значение для принудительно задаваемого входа или выхода в строке "Принудительное значение". После можно установить флажок в столбце "Принудительное присваивание значений", чтобы активировать принудительную установку значений входа или выхода.



Нажать на кнопку "Запустить или заменить принудительное присваивание значений", чтобы установить принудительные значения для переменных (тегов) в таблице принудительно задаваемых значений. Нажать на кнопку "Завершить принудительное присваивание значений", чтобы сбросить значения для переменных (тегов).

После в таблице принудительно задаваемых значений можно отслеживать состояние принудительно установленного значения для входа. Но мониторинг принудительно установленного значения для выхода невозможен.

Можно отображать принудительно установленные значения и в редакторе текстов программы.



Примечание

Когда вход или выход принудительно прописывается в таблице принудительно задаваемых значений, операции принудительной установки становятся частью конфигурации проекта. После закрытия STEP 7, принудительно установленные элементы остаются активными в программе CPU, пока они не будут сброшены. Чтобы сбросить эти принудительные элементы, следует использовать STEP 7 для подключения к Онлайн-CPU, а затем таблицу принудительно задаваемых значений для выключения или остановки функции принудительной установки для этих элементов.

15.13.5.2 Принцип работы функции принудительной установки

CPU позволяет принудительно устанавливать входы и выходы, указывая физический адрес входа или выхода (I_:P или Q_:P) в таблице принудительно задаваемых значений и запуская после функцию принудительной установки.

В программе считанные значения физических входов заменяются принудительным значением. Программа использует принудительное значение при обработке. Когда программа пишет на физический выход, выходное значение заменяется принудительным значением. Принудительное значение появляется на физическом выходе и используется процессом.

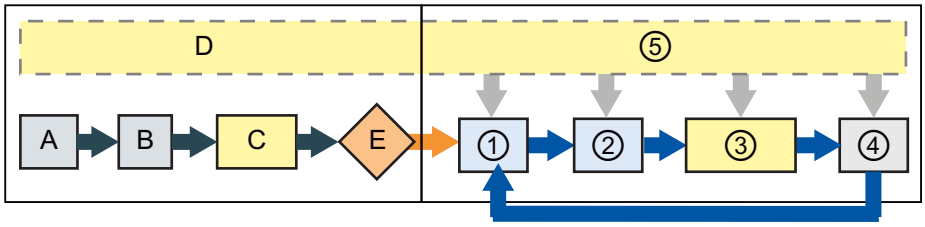
Когда вход или выход принудительно прописывается в таблице принудительно задаваемых значений, операции принудительной установки становятся частью программы пользователя. Даже при условии, что среда программирования была закрыта, принудительные значения остаются активными в работающей программе CPU, пока они не будут сброшены при переходе онлайн с помощью среды программирования, путем остановки функции принудительной установки. Программы с принудительно установленными входами и выходами, после загрузки в другой CPU из карты памяти, будут продолжать принудительно устанавливать выбранные в программе входы и выходы.

Если CPU выполняет пользовательскую программу с карты памяти, защищенной от записи, нельзя запустить или изменить принудительную установку ввода-вывода из таблицы текущего контроля, потому что нельзя переопределить значения в пользовательской программе, защищенной от записи. Любая попытка принудительной установки значений, защищенных от записи, генерирует ошибку. При использовании карты памяти для переноса пользовательской программы, любые принудительно установленные элементы на этой карте памяти будут переданы в CPU.

Примечание

Принудительная активация назначенных HSC, PWM и PTO цифровых I/O невозможна

Используемые высокоскоростными счётчиками (HSC), широтно-импульсной модуляцией (PWM) и выводом последовательности импульсов (PTO) цифровые I/O назначаются при конфигурировании устройства. Если этим функциям назначаются цифровые I/O, то значения адресов назначенных I/O не могут быть изменены с помощью функции принудительной установки значений в таблице принудительно задаваемых значений.



Пуск

A Очистка I области памяти не затрагивается функцией принудительной установки.

B Инициализация выходных значений не затрагивается функцией принудительной установки.

C В процессе выполнения пускового OB, CPU применяет принудительное значение, когда программа пользователя обращается к физическому входу.

D Сохранение событий прерывания в листе ожидания не затрагивается.

E Разрешение записи в выходы не затрагивается.

RUN

① При записи Q памяти на физические выходы CPU применяет принудительные значения при обновлении выходов.

② При чтении физических входов CPU применяет принудительные значения как раз перед копированием входов в I память.

③ В процессе выполнения программы пользователя (OB программного цикла), CPU применяет принудительное значение, когда программа пользователя обращается к физическому входу или выполняет запись на физический выход.

④ Обработка коммуникационных запросов и самодиагностика не затрагиваются функцией принудительной установки.

⑤ Обработка прерываний во время любого этапа цикла не затрагивается.

15.14 Загрузка в рабочем состоянии RUN

CPU поддерживает "Загрузку в рабочем состоянии RUN". Эта функция позволяет вносить небольшие изменения в пользовательскую программу без воздействия на процесс, управляемый этой программой. Но использование этой функции позволяет вносить и серьезные изменения в программу, что может стать дезорганизующим или даже опасным для процесса.

⚠ ПРЕДУПРЕЖДЕНИЕ

Риски при загрузке в рабочем состоянии RUN

При загрузке изменения в CPU в режиме RUN, они сразу влияют на работу процесса. Изменение программы в режиме RUN может привести к непредсказуемой работе системы, что в свою очередь может стать причиной смерти или серьезных травм персонала и/или повреждения оборудования.

Только уполномоченный лица, которые осознают последствия от внесения изменений в режиме RUN на работу системы, могут выполнить загрузку в режиме RUN.

Функция "Загрузка в рабочем состоянии RUN" позволяет вносить изменения в программу и загружать их в CPU, не переключаясь в режим STOP.

- Можно внести незначительные изменения в свой работающий процесс, не останавливая его (например, изменить значение параметра).
- Кроме этого, с помощью этой функцией можно выполнить отладку программы быстрее (например, инвертировать логику для NO- или NC-контакта).

Можно вносить следующие изменения в программные блоки и переменные (теги) с загрузкой их в режиме RUN:

- Создание, перезапись и удаление функций (FC), функциональных блоков (FB) и таблиц переменных (тегов).
- Создание, удаление и перезапись блоков данных (DB) и экземплярных блоков данных для функциональных блоков (FB). Можно добавить элементы в структуры DB и загрузить их в режиме RUN. CPU может сохранить значения существующих переменных (тегов) блока и инициализировать новые переменные (теги) блока данных согласно начальным значениям, или CPU может установить все переменные (теги) блока данных на начальные значения, в зависимости от настроек конфигурации (Страница 1416). Нельзя загрузить DB веб-сервера (управление или фрагмент) в режиме RUN.
- Перезапись организационных блоков (OB); однако, нельзя создать или удалить OB.

Можно загрузить максимум двадцать блоков в режиме RUN за один раз. Если необходимо загрузить больше двадцати блоков, то следует перевести CPU в режим STOP.

При загрузке изменений в реальный процесс (в противоположность моделируемому процессу, что можно делать в ходе отладки программы), крайне важно продумать с точки зрения безопасности возможные последствия для машин и их операторов, прежде чем выполнить загрузку.

Примечание

Если CPU находится в режиме RUN, и в программу были внесены изменения, STEP 7 всегда пытается вначале загрузить изменения в RUN. Если это не нужно, то следует перевести CPU в состояние STOP.

Если внесенные изменения не поддерживаются функцией "Загрузка в рабочем состоянии RUN", STEP 7 предлагает пользователю перевести CPU в STOP.

15.14.1 Условия для "Загрузки в рабочем состоянии RUN"

Возможность для загрузки программных изменений в CPU, который находится в режиме RUN, предполагает наличие следующих условий:

- CPU должен иметь версию V3.0 или выше

  Примечание

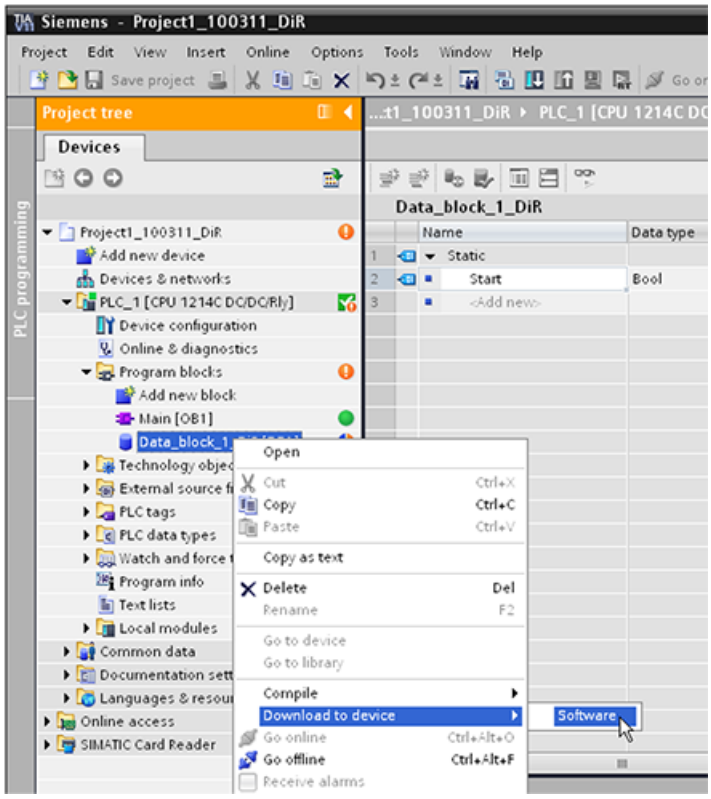
  Версия CPU должна быть не ниже V4.0, чтобы изменить расширенный интерфейс блока в режиме RUN (Страница 1416).

- Программа должна быть успешно скомпилирована.
- Необходимо наличие успешно установленного соединения между программатором, на котором работает STEP 7, и CPU.

15.14.2 Редактирование программы в рабочем состоянии RUN

Чтобы изменить программу в режиме RUN, вначале следует убедиться, что CPU и программа удовлетворяют необходимым условиям (Страница 1412), и затем выполнить следующие действия:

1. Чтобы загрузить программу в режиме RUN, действуйте согласно одному из следующих методов:
   - Выбрать команду "Загрузка в устройство" из меню "Онлайн".
   - Нажать кнопку "Загрузка в устройство" на панели инструментов.
   - В "дереве проекта" кликнуть правой кнопкой по "Программные блоки" и выбрать команду "Загрузка в устройство > ПО".



   Если программа компилируется успешно, то STEP 7 начинает загружать программу в CPU.

2. Когда STEP 7 предложит загрузить программу или отменить операцию, нажать "Загрузить", чтобы загрузить программу в CPU.

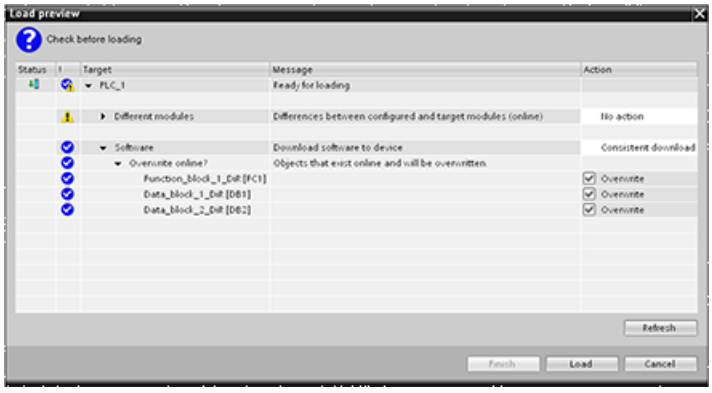
15.14.3 Загрузка выбранных блоков

Из папки "Программные блоки" можно выбрать отдельный блок или несколько блоков для загрузки.

Если выбирается отдельный блок для загрузки, то единственной опцией в столбце "Действие" является "Согласованная загрузка". Можно развернуть строку категории, чтобы проверить, какие блоки должны быть загружены. В данном примере были внесены незначительные изменения в офлайн блоке, и никакие другие блоки не должны быть загружены.



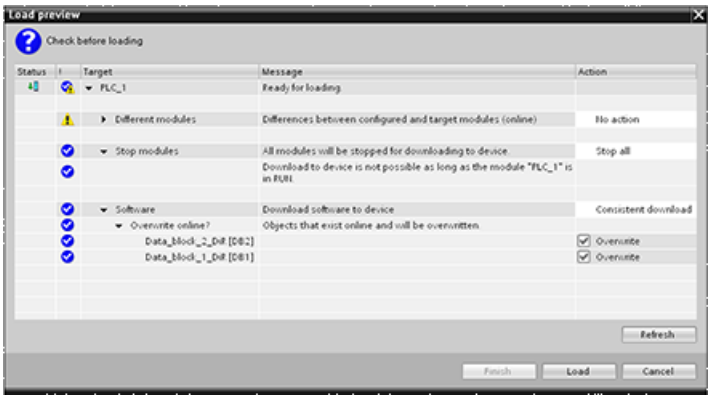
В данном примере необходима загрузка нескольких блоков.



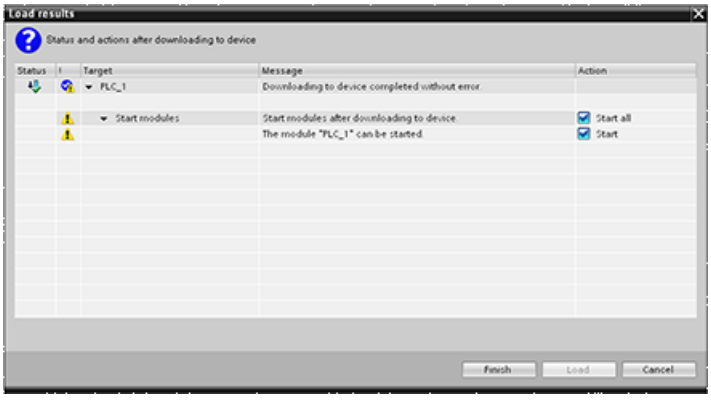
Примечание

Можно загрузить максимум двадцать блоков в режиме RUN за один раз. Если необходимо загрузить больше двадцати блоков, то следует перевести CPU в режим STOP.

При попытке выполнить загрузку блоков в RUN, система обнаруживает до фактической загрузки, что это невозможно, в диалоговом окне появляется строка для остановки модулей.

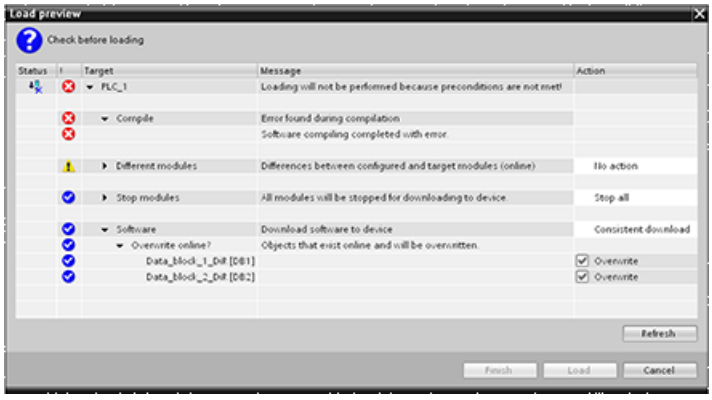


Нажать на кнопку "Загрузить". Откроется диалоговое окно "Результаты загрузки". Для завершения процесса загрузки нажать на экранную кнопку "Завершить".



15.14.4 Загрузка одиночного выбранного блока с ошибкой компиляции в другом блоке

Если попытаться выполнить согласованную загрузку с ошибкой компиляции в другом блоке, то диалоговое окно указывает на ошибку, и кнопка загрузки неактивна.



Сначала необходимо исправить ошибку компиляции в другом блоке. После этого кнопка "Загрузить" становится активной.



15.14.5 Изменение и загрузка существующих блоков в целевую систему в рабочем состоянии RUN

Функция для загрузки в рабочем состоянии RUN позволяет добавлять и изменять переменные (теги) в блоках данных и функциональных блоках и затем загружать измененный блок в CPU в рабочем состоянии RUN.

Загрузка без повторной инициализации

У каждого DB и FB есть резервный объем памяти, которую можно использовать для добавления переменных (тегов) в блок, который впоследствии можно загрузить в режиме RUN. По умолчанию начальный размер резерва памяти составляет 100 байтов. Можно добавлять дополнительные переменные (теги) к данным с размером не более, чем резерв памяти, и загружать расширенный блок в CPU в режиме RUN. Также можно увеличить резерв, если требуется больше памяти для дополнительных переменных (тегов) в блоке. Если добавляется больше переменных (тегов), чем выделенный объем памяти, то нельзя загрузить расширенный блок в CPU в режиме RUN.



Функция "Загрузка без повторной инициализации" позволяет расширять блок данных, добавляя больше переменных (тегов) блока данных и загружать расширенный блок данных в режиме RUN в CPU. В этом случае можно добавить переменные (теги) в блок данных и загрузить его, не выполняя повторную инициализацию в программе. CPU

сохраняет значения существующих переменных (тегов) блока данных и инициализирует вновь добавленные переменные (теги) согласно их стартовым значениям.

Чтобы разрешить эту функцию для онлайн проекта с CPU в режиме RUN, выполнить следующие шаги:

1. Открыть блок в папке программных блоков в дереве проекта STEP 7.
2. Нажать на кнопку "Загрузка без повторной инициализации" в редакторе блоков, чтобы разрешить функцию. (Значок в активном состоянии выделен рамкой:)
3. Подтвердить выбор, нажав "ОК".
4. Добавить переменные (теги) в интерфейс блока и загрузить блок в режиме RUN. Можно добавить и загрузить столько новых переменных (тегов), сколько позволяет резерв памяти.

Если было добавлено больше байтов к блоку, чем было сконфигурировано для резерва памяти, STEP 7 показывает ошибку, как только будет предпринята попытка загрузить блок в режиме RUN. Необходимо отредактировать свойства блока и увеличить резерв памяти. Не следует удалять существующие записи или изменять резерв памяти блока, пока включена функция “Загрузка без повторной инициализации”. Чтобы отключить "Загрузку без повторной инициализации" функция, выполнить следующие шаги:

1. Нажать на кнопку "Загрузка без повторной инициализации" в редакторе блоков, чтобы выключить функцию. (Значок в не активном состоянии не выделен рамкой:)
2. Подтвердить выбор, нажав "ОК".
3. Загрузить блок в CPU. В диалоговом окне загрузки следует выбрать "Повторная инициализация", чтобы загрузить расширенный блок.

Загрузка в этом случае повторно инициализирует все существующие и новые переменные (теги) блока согласно их стартовым значениям.

Загрузка сохраняемых переменных (тегов) блока

Загрузка сохраняемых переменных (тегов) блока в режиме RUN требует выделения резерва сохраняемой памяти. Чтобы сконфигурировать этот резерв сохраняемой памяти, выполнить следующие действия:

1. В папке программных блоков в дереве проекта STEP 7 кликнуть правой кнопкой по блоку и выбрать "Свойства" из контекстного меню.
2. Выбрать свойство "Загрузка без повторной инициализации".
3. Установить флажок для "Разрешить загрузку без повторной инициализации для сохраняемых переменных (тегов)".
4. Сконфигурировать число байтов, доступных для резерва сохраняемой памяти.
5. Подтвердить изменения, нажав "ОК".
6. Добавить сохраняемые переменные (теги) в блок данных и загрузить блок данных в режиме RUN. Можно добавить и загрузить столько новых сохраняемых переменных (тегов) блока данных, сколько позволяет резерв сохраняющей памяти.

Если было добавлено больше сохраняемых байтов к блоку данных, чем было сконфигурировано для резерва памяти, STEP 7 показывает ошибку, как только будет предпринята попытка загрузить блок в режиме RUN. Можно добавить сохраняемые

переменные (теги) блока объемом не более резерва сохраняющей памяти, чтобы позволить их загрузку в режиме RUN.

При загрузке дополнительные сохраняемых переменных (тегов) блока, переменные (теги) содержат свои текущие значения.

Конфигурирование размера резерва памяти для новых блоков

Объем резерва памяти по умолчанию для новых блоков данных составляет 100 байтов. При создании нового блока, он имеет в наличии 100 резервных байтов. Если необходимо, чтобы размер резерва памяти для новых блоков был другим, можно изменить установки в настройках программирования PLC:

1. Выбрать в STEP 7 команду меню **Опции > Настройки**.
2. В диалоговом окне "Настройки" развернуть область "Программирование PLC" и выбрать "Общее".
3. В разделе "Загрузка без повторной инициализации" ввести число байтов для резерва памяти.

При создании новых блоков STEP 7 будет использовать резерв памяти, сконфигурированный для новых блоков.

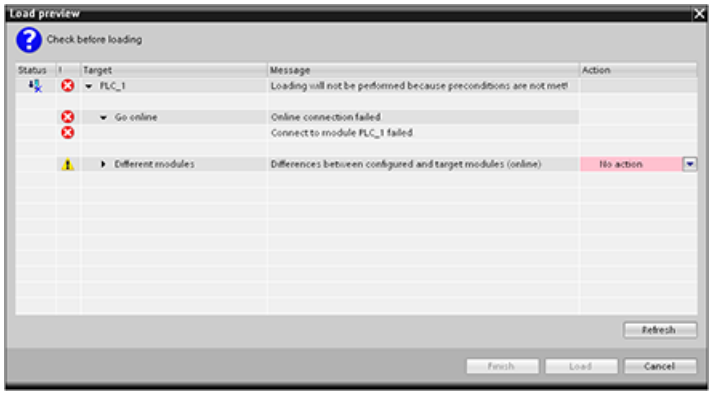
Ограничения

Следующие ограничения действуют для редактирования и загрузки блоков в режиме RUN:

- Расширение интерфейса блока, добавление новых переменных (тегов) и загрузка в режиме RUN доступны только для оптимизированных блоков (Страница 186).
- Нельзя изменить структуру блока и загрузить измененный блок в режиме RUN без повторной инициализации. Добавление новых элементов в переменную (тег) с типом Struct (Страница 132), изменение имен переменных (тегов), размеров массивов, типов данных или сохраняемого состояния требуют, чтобы блок был повторно инициализирован при загрузке его в режиме RUN. Единственными изменениями существующих переменных (тегов) блока, которые можно выполнить, и загрузить блок в режиме RUN без повторной инициализации, являются изменения стартовых значений (блоки данных), значений по умолчанию (функциональные блоки) или комментарии.
- Нельзя загрузить больше новых переменных (тегов) блока в режиме RUN, чем позволяет разместить резерв памяти.
- Нельзя загрузить больше новых сохраняемых переменных (тегов) блока в режиме RUN, чем позволяет разместить резерв сохраняющей памяти.

15.14.6 Реакция системы на неудачный процесс загрузки

Во время первой загрузки в режиме RUN, если происходит сбой сетевого соединения, STEP 7 выводит на экран следующее диалоговое окно "Предпросмотр загрузки":



15.14.7 Критерии безопасности при загрузке в рабочем состоянии RUN

Прежде, чем загрузить программу в рабочем состоянии RUN, следует оценить влияние изменений в режиме RUN на работу CPU для следующих ситуаций:

- Если удалить управляющую логику для выхода, CPU сохраняет последнее состояние выхода до следующего цикла включения и выключения питания или перехода в режим STOP.
- Если удалить высокоскоростной счетчик или импульсный выход, которые работали, то они продолжают работать до следующего цикла включения и выключения питания или перехода в режим STOP.

- Логика, которая активируется меркером первого цикла, не будет выполняться до следующего цикла включения и выключения питания или перехода из STOP в RUN. Меркер первого цикла устанавливается только при переходе в режим RUN и загрузка в режиме RUN не оказывает на него никакого воздействия.
- Текущие значения блоков данных (DB) и/или переменных (тегов) могут быть перезаписаны.

Примечание

Для загрузки программы в рабочем состоянии RUN, CPU должен поддерживать загрузку в режиме RUN. Программа должна быть скомпилироваться без ошибок, и соединение между STEP 7 и CPU должно быть устойчивым.

Можно вносить следующие изменения в программные блоки и переменные (теги) с загрузкой их в режиме RUN:

- Создание, перезапись и удаление функций (FC), функциональных блоков (FB) и таблиц переменных (тегов).
- Создание и удаление блоков данных (DB); однако, изменения структуры DB не могут быть перезаписаны. Могут быть перезаписаны начальные значения DB. Нельзя загрузить DB веб-сервера (управление или фрагмент) в режиме RUN.
- Перезапись организационных блоков (OB); однако, нельзя создать или удалить OB.

Можно загрузить максимум двадцать блоков в режиме RUN за один раз. Если необходимо загрузить больше двадцати блоков, то следует перевести CPU в режим STOP.

После запуска загрузки, другие задачи в STEP 7 не могут выполняться, пока загрузка не завершится.

Инструкции, которые могут перестать работать при "Загрузке в рабочем состоянии RUN"

Следующие инструкции могли бы столкнуться с нерегулярной ошибкой, когда в CPU активируются изменения после загрузки в режиме RUN. Ошибка происходит, когда инструкция вызывается, в то время как CPU готовится активировать загруженные изменения. В течение этого времени CPU приостанавливает дальнейший доступ пользовательской программы к загружаемой памяти, пока он завершает активный доступ пользовательской программы к загружаемой памяти. Это делается для того, чтобы загруженные изменения могли быть активированы согласованно.

Инструкция	Ответ во время ожидания активации
DataLogCreate	STATUS = W#16#80C0, ERROR = WAHR
DataLogOpen	STATUS = W#16#80C0, ERROR = WAHR
DataLogWrite	STATUS = W#16#80C0, ERROR = WAHR
DataLogClose	STATUS = W#16#80C0, ERROR = WAHR
DataLogNewFile	STATUS = W#16#80C0, ERROR = WAHR
DataLogClear	STATUS = W#16#80C0, ERROR = WAHR
DataLogDelete	STATUS = W#16#80C0, ERROR = WAHR
READ_DBL	RET_VAL = W#16#82C0
WRIT_DBL	RET_VAL = W#16#82C0
Create_DB	RET_VAL = W#16#80C0

Инструкция	Ответ во время ожидания активации
Delete_DB	RET_VAL = W#16#80C0
RTM	RET_VAL = 0x80C0

Во всех случаях RLO инструкции будет иметь значение false, когда происходит ошибка. Ошибка является временной. Если это происходит, инструкция должна быть повторена позже.

Примечание

Не следует повторять инструкцию в текущем исполнении OB.

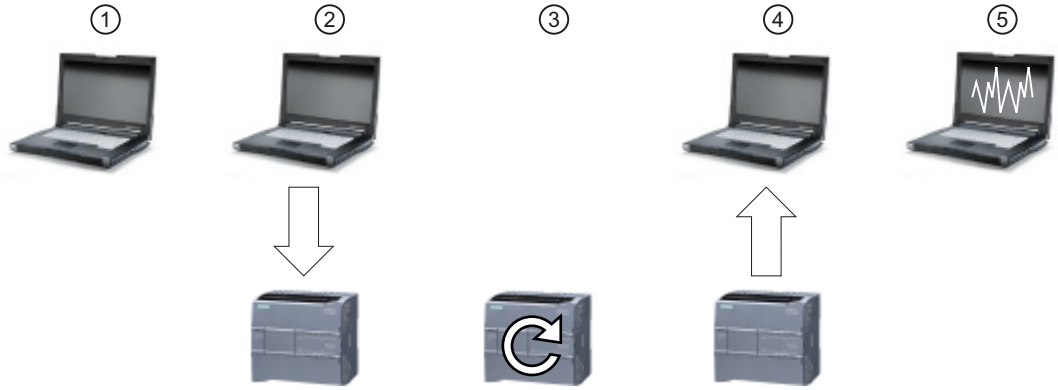
15.15 Трассировка и запись данных CPU по условиям запуска

STEP 7 предлагает функции трассировки и анализатора логики, с помощью которых можно сконфигурировать переменные для отслеживания и записи со стороны PLC. После можно загрузить записанные данные измерений в программатор и использовать инструменты STEP 7, чтобы анализировать, обрабатывать и представлять данные в виде графиков. В папке "Трассировки" в дереве проекта STEP 7 можно создавать и управлять трассировками.

Примечание

Данные трассировок доступны только в проекте STEP 7 и недоступны для обработки другими инструментами.

На изображении ниже показан ход процесса функции трассировки:



① Сконфигурировать функцию трассировки в редакторе трассировки STEP 7. Можно определить следующие опции:

- Значения данных для записи
- Продолжительность записи
- Интервал записи
- Условие запуска

② Передать конфигурацию трассировки из STEP 7 в PLC.

③ PLC выполняет программу, и когда выполняется условие запуска, начинает записывать данные трассировки.

④ Передать записанные значения из PLC в STEP 7.

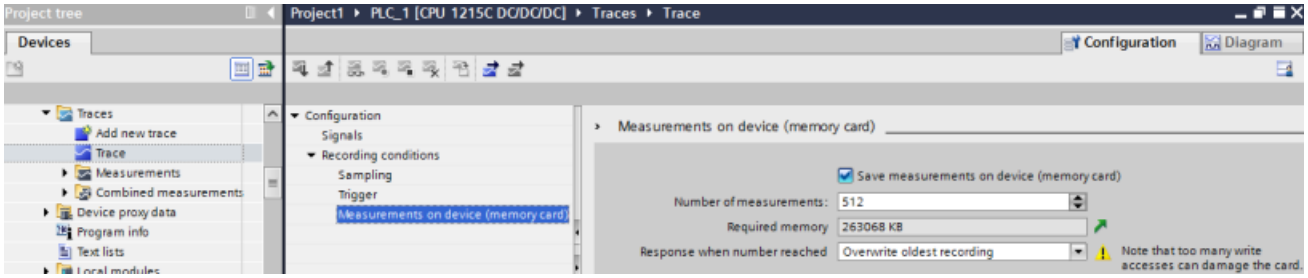
⑤ Используя инструменты в STEP 7, проанализировать данные, представить их графически и сохранить.

S7-1200 поддерживает две операции трассировки максимум с 16 регистрируемыми переменными на событие трассировки. Каждая операция трассировки занимает 524288 байтов оперативной памяти для записи значений трассировки и связанных служебных данных, например, адресов переменных и отметок времени.

Сохранение результатов измерений на карту памяти

S7-1200 CPU может сохранять результаты измерений только на карту памяти SIMATIC. Если в CPU нет карты памяти, то CPU выполнят запись в буфер диагностики при попытке со стороны программы сохранить результаты измерений. CPU ограничивает пространство, выделенное для результатов трассировки, так что всегда должен быть доступен 1 МБ внешней загружаемой памяти. Если для результатов трассировки потребуется больше памяти, чем максимальное допустимое значение, CPU не сохранит измерение и внесет запись в буфер диагностики.

Кроме того, если выбрать "Заменять самую старую запись" в STEP 7, непрерывная запись может сократить срок службы загружаемой памяти. При выборе "Заменять самую старую запись", CPU заменяет самое старое измерение самым последним измерением после сохранения сконфигурированного числа результатов измерений. После следующие результаты измерения записываются и сохраняются. Перезапись самых старых измерений полезна при диагностике единичных проблем.



CPU поддерживает максимум 999 результатов измерений. Во время сохранения результатов трассировки во внешнюю загружаемую память, CPU не проверяет условие запуска для операции трассировки. Как только CPU завершит сохранение результатов измерений, он возобновляет проверку условий запуска.

Доступ к примерам

В информационной системе STEP 7 можно найти дополнительную информацию о том, как программировать трассировку, как загрузить конфигурацию, загрузить данные трассировки и отобразить данные в анализаторе логики. Подробные примеры в главе "Использование онлайн- и диагностических функций > Использование функции трассировки и анализатора логики".

Кроме этого полезной помощью является онлайн руководство "Промышленная Автоматизация SINAMICS/SIMATIC. Использование функции трассировки и анализатора логики" (https://support.industry.siemens.com/cs/ww/en/view/64897128)

15.16 Определение типа обрыва провода с помощью модуля SM 1231

Как описано в Диапазоны измерений аналоговых входов для напряжения и тока (SB и SM) (Страница 1548), модуль SM 1231 возвращает аналоговое входное значение 32767 (16 # 7FFF) как при обрыве провода, так и при переполнении. Если необходимо определить, какое из этих двух условий имело место, то можно включить логику для определения в программу STEP 7. Метод определения состояния состоит из следующих задач:

- Создать OB диагностического прерывания, который вызывается при наличии наступающего или уходящего события диагностики.
- Вставить вызов инструкции RALRM.
- Установить массив из байт для параметра AINFO, который содержит информацию о состоянии.
- Проанализировать байты 32 и 33 структуры AINFO из RALRM_DB, когда CPU запустит OB диагностического прерывания.

Создание OB диагностического прерывания

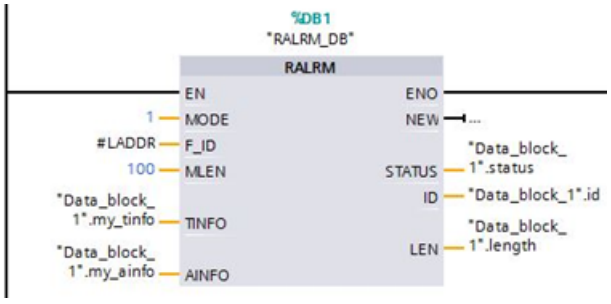
Чтобы определить, когда происходит обрыв провода, необходимо создать OB диагностического прерывания. CPU будет вызывать этот OB всякий раз при наличии наступающего или уходящего события диагностики.

Когда CPU вызывает OB диагностического прерывания, входной параметр LADDR содержит аппаратный идентификатор модуля с ошибкой. Аппаратный идентификатор для модуля SM 1231 находится в конфигурации устройства STEP 7 модуля SM 1231.

Вызов инструкции RALRM

Для программирования вызова инструкции RALRM выполнить следующие действия:

1. Вставить вызов RALRM в программу STEP 7.
2. Установить для входного параметра F_ID аппаратный идентификатора параметра LADDR для OB диагностического прерывания.
3. Использовать массив из байт для входных параметров TINFO и AINFO. Использовать массив с размером не менее 34 байт.



Анализ AINFO после возникновения диагностического прерывания

После выполнения OB диагностического прерывания, байтовый массив AINFO содержит информацию о диагностике модуля.

Байт от 0 до 25 - это информация заголовка. Следующие байты относятся к диагностике модуля:

Байт	Описание	
26 и 27	Значение слова 16#8000 - указывает на то, что речь идет о диагностике в стиле PROFINET	
28 и 29	Слово, содержащее номер отвечающего за диагностику канала	
30	Битовая комбинация aaabb000 указывает на тип канала (aaa) и тип ошибки (bb)	
	aaa	bb
	000: Зарезервировано	00: Зарезервировано
	001: Входной канал	01: Входящая ошибка
	010: Выходной канал	10: Исходящая ошибка
	011: Входной/выходной канал	11: Исходящая ошибка, имеются другие ошибки

Байт	Описание
31	Отображение формата данных 0: Свободный формат данных 1: Бит 2: Два бита 3: Четыре бита 4: Байт 5: Слово (два байта) 6: Двойное слово (четыре байта) 7: Два двойных слова (восемь байт)
32 и 33	Слово, определяющее тип ошибки 16#0000: зарезервировано 16#0001: короткое замыкание 16#0002: напряжение ниже допустимого 16#0003: напряжение выше допустимого 16#0004: перегрузка 16#0005: перегрев 16#0006: обрыв провода 16#0007: выход за верхний предел 16#0008: выход за нижний предел 16#0009: ошибка

К примеру, рассмотрим байты от 26 до 33 этой структуры AINFO:

29		my_ainfo[26]	Byte	16#0	16#80
30		my_ainfo[27]	Byte	16#0	16#00
31		my_ainfo[28]	Byte	16#0	16#00
32		my_ainfo[29]	Byte	16#0	16#00
33		my_ainfo[30]	Byte	16#0	16#28
34		my_ainfo[31]	Byte	16#0	16#05
35		my_ainfo[32]	Byte	16#0	16#00
36		my_ainfo[33]	Byte	16#0	16#07

- Слово 16#8000 в байтах 26 и 27 указывает на то, что речь идет о диагностике в стиле PROFINET.
- Слово в байтах 28 и 29 показывает, что это диагностика для канала 0 или модуля.
- Байт 30 =16#28, который интерпретируется при анализе как битовая комбинация aaa bb 00 = 001 01 000. Значение указывает на то, что эта диагностика относится к входному каналу и что речь идет о входящей ошибке.
- Байт 31 = 5, что указывает на значение типа слово.
- Значение слова в байтах 32 и 33 = 16#0007, что указывается на выход за верхний предел.

Путем сбора информации AINFO из события диагностического прерывания, можно определить причину диагностического события.

15.17 Резервное копирование и восстановление CPU

15.17.1 Опции для резервного копирования и восстановления

С течением времени в систему автоматизации вносятся изменения, например, добавляются новые устройства, заменяются существующие устройства или редактируется пользовательская программа. Если такие изменения приводят к нежелательному поведению, можно восстановить систему автоматизации в более ранней версии, если для нее была создана резервная копия. STEP 7 и S7-1200 CPU предлагают различные опции для резервного копирования и восстановления конфигурации оборудования и программного обеспечения.

Опции для резервного копирования

В следующей таблице представлен обзор опций для резервного копирования и восстановления S7-CPU:

	Мгновенный снимок контролируемых значений	**Загрузка из устройства (ПО)**	**Выгрузка устройства как новой станции (оборудование и ПО)**	**Загрузка резервной копии из онлайн-устройства**
Случай применения	Восстановление определенного состояния для блока данных. Фактические значения блоков данных с отметкой времени включаются в проект.	Выгрузка блоков из CPU в проект.	Выгрузка конфигурации оборудования и ПО из устройства в проект.	Создание полной резервной копии CPU как точки восстановления. Резервная копия является согласованной и не может быть изменена или открыта.
Необходимое условие	CPU существует в проекте. Блоки данных онлайн и офлайн должны совпадать.	CPU существует в проекте.	Устройство присутствует в каталоге оборудования TIA Portal. Все необходимые HSP или файлы GSD установлены.	-
Возможно в рабочем состоянии	RUN, STOP	RUN, STOP	RUN, STOP	STOP
Возможно для F-CPU	Да	Да	Нет	Да
Возможность редактирование резервной копии	Да	Да	Да	Нет

Содержимое резервной копии

В следующей таблице показано, какие данные могут быть загружены для создания резервных копий и какие опции для этого потребуются:

	Мгновенный снимок контролируемых значений	**Загрузка из устройства (ПО)**	**Выгрузка устройства как новой станции (оборудование и ПО)**	**Загрузка резервной копии из онлайн-устройства**
Фактические значения блоков данных	Мгновенный снимок возможен	Загрузка возможна	Загрузка возможна	Резервное копирование возможно
Программные блоки	-	Загрузка возможна	Загрузка возможна	Резервное копирование возможно
Переменные (теги) PLC (имена переменных (тегов) и констант)	-	Загрузка возможна	Загрузка возможна	Резервное копирование возможно
Технологические объекты	-	Загрузка возможна	Загрузка возможна	Резервное копирование возможно
Конфигурация оборудования	-	-	Загрузка возможна	Резервное копирование возможно
Таблицы текущего контроля (веб-сервер)	-	-	Загрузка невозможна	Резервное копирование возможно
Локальные данные, битовая память (меркеры), таймеры, счетчик и образ процесса	Мгновенный снимок невозможен	Загрузка невозможна	Загрузка невозможна	Резервное копирование возможно
Архивы и рецепты (PLC)	-	-	-	Резервное копирование возможно
Общие данные на карте памяти SIMATIC, например, справка для программных блоков или файлов GSD	-	-	-	Резервное копирование возможно

Особенности резервного копирования для фактических значений

Тип резервной копии "Загрузка резервной копии из онлайн-устройства" содержит фактические значений переменных (тегов), которые определены как сохраняющиеся. Для обеспечения непротиворечивости сохраняющихся данных, следует отключить все обращения по записи к сохраняющимся данным во время резервного копирования.

Переход из режима STOP в режим RUN устанавливает фактические значения не сохраняющихся данных на их стартовые значения. Резервная копия CPU содержит только стартовые значения не сохраняющихся данных.

15.17.2 Резервное копирование Онлайн-CPU

Создание резервной копии текущей конфигурации имеет смысл, если в будущем планируется возврат к определенной конфигурации. В этом случае можно будет восстановить текущую конфигурацию.

Необходимое условие

Можно создать любое число резервных копий, храня различные конфигурации для CPU. Для создания резервной копии должны быть выполнены следующие условия:

- CPU в проекте STEP 7 уже создан.
- CPU подключен через PROFINET интерфейс напрямую к программатору/PC. Операции резервного копирования и восстановления не поддерживают PROFIBUS интерфейсов на CM.
- CPU находится в режиме online. (При отсутствии Online-соединения, процесс резервного копирования создает Online-соединение.)
- CPU находится в рабочем состоянии STOP. (Если CPU не находится в рабочем состоянии STOP, то процесс резервного копирования просит подтвердить, что CPU может быть переведен в режим STOP.)

Порядок действий

Для создания резервной копии из текущей конфигурации CPU действовать следующим образом:

1. Выбрать CPU в дереве проекта.
2. Выбрать в меню "Онлайн" команду "Загрузка резервной копии из онлайн-устройства". Может потребоваться ввод пароля для доступа по чтению к CPU и подтверждение, что CPU должен перейти в рабочее состояние STOP.

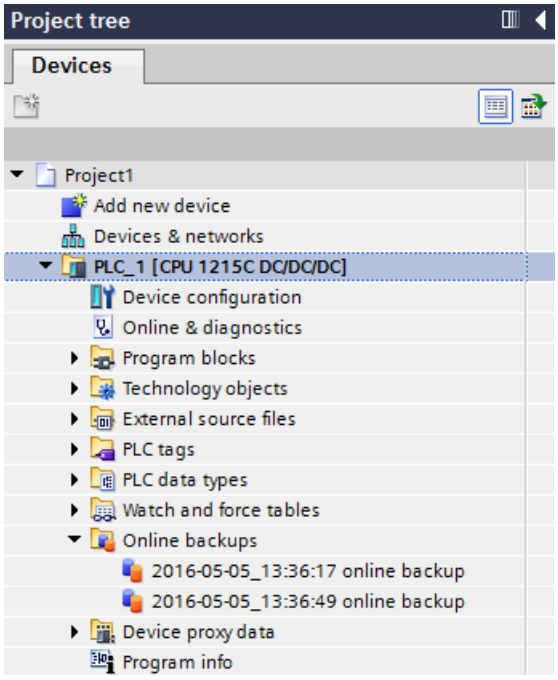
Результат

Присвоение имени для резервной копии производится по имени CPU, а также времени и дате резервного копирования. Резервная копия содержи все данные, необходимые для восстановления определенной конфигурации CPU. CPU включает в резервную копию следующие данные:

- Содержимое карты памяти, при наличии
- Сохраняющие области памяти блоков данных, счетчиков и меркеров
- Другое сохраняющееся содержимое памяти, напр., параметры IP-адреса

Резервная копия содержит фактические значения CPU, но не буфер диагностики.

Резервные копии можно найти в дереве проекта в разделе CPU в папке "Резервные онлайн-копии". На следующем рисунке показан S7-1200 CPU, для которого были созданы две резервные копии:



Примечание

Обратить внимание, что также можно создать резервную копию Онлайн-CPU с помощью SIMATIC Automation Tool (SAT) или стандартной веб-страницы "Резервное онлайн-копирование" веб-сервера (Страница 1059).

При резервном копировании файлов через STEP 7 они сохраняются в проект STEP 7. При резервном копировании файлов через веб-сервер, PC или устройство сохраняет резервные файлы в стандартную парку для загрузки. Нельзя восстановить резервные файлы STEP 7 через веб-сервер, как нельзя восстановить и резервные файлы веб-сервера через STEP 7. Но можно сохранить резервные файлы STEP 7 напрямую в папку загрузки PC или устройства. Если действовать таким образом, то файлы смогут быть восстановлены через веб-сервер.

Сохранение резервных файлов на PC или устройстве

Для сохранения резервного файла на своем PC или устройстве необходимо выполнить следующее:

1. В дереве проекта кликнуть правой кнопкой мыши по файлу в папке Резервные онлайн-копии.
2. Выбрать в контекстном меню "Сохранить как".
3. Перейти к папке, в которую необходимо сохранить файл, напр., к стандартной папке для загрузок на PC или устройстве.
4. Кликнуть по "Сохранить".

15.17.3 Восстановление CPU

Если резервная копия конфигурации CPU была создана ранее, то ее можно снова перенести на CPU. При восстановлении из резервной копии CPU переходит в рабочее состояние STOP. Если для CPU сконфигурирован уровень доступа, то необходимо ввести пароль для доступа к CPU по чтению.

⚠ ПРЕДУПРЕЖДЕНИЕ

Восстановление резервных копий с неизвестными содержимым

Восстановление резервной копии с неизвестным содержимым может нанести серьезный материальный ущерб или вред здоровью в случае неисправностей или программных ошибок.

Следует убедиться, что в резервную копию включена конфигурация с известным содержимым.

Необходимое условие

Для восстановления резервной копии должны быть выполнены следующие условия:

- В проекте STEP 7 есть конфигурация для CPU и созданная ранее резервная копия.
- CPU подключен через PROFINET интерфейс CPU напрямую к программатору/PC.
- CPU находится в рабочем состоянии STOP.
- Если сконфигурирован уровень доступа, то необходимо знать пароль для полного доступа к CPU.

Порядок действий

Для восстановления резервной копии выполнить следующие действия:

1. Открыть CPU в дереве проекта, чтобы показать подчиненные объекты.
2. Выбрать в папке "Резервные онлайн-копии" резервную копию, которую необходимо восстановить.
3. Выбрать команду "Загрузка в устройство" из меню "Онлайн".
   - Если прежде уже было установлено онлайн-соединение (Страница 1390), то отображается диалоговое окно "Предпросмотр загрузки". В этом диалоговом окне отображаются предупреждения и рекомендуемые действия для процесса загрузки.
   - Если онлайн-соединение еще не было установлено, откроется диалоговое окно "Расширенная загрузка в устройство" и сначала необходимо будет выбрать интерфейс, через который следует установить онлайн-соединение с CPU.
4. Проверить предупреждения в диалоговом окне "Предпросмотр загрузки" и при необходимости выбрать действия в столбце "Действие".
5. Нажать на кнопку "Загрузить". (Когда загрузка будет возможной, кнопка "Загрузка" станет активной.)

6. STEP 7 восстанавливает резервную копию в CPU. В диалоговом окне "Результаты процесса загрузки" можно проверить, насколько успешной была загрузка, и при необходимости выбрать дальнейшие действия.
7. После проверки в диалоговом окне "Результаты процесса загрузки" нажать на кнопку "Завершить".
   Может потребоваться ввод пароля для полного доступа к CPU и подтверждение, что CPU должен перейти в рабочее состояние STOP.
   STEP 7 восстанавливает содержимое резервной копии в CPU и перезапускает CPU.

Примечание

Обратить внимание, что можно восстановить резервную копию и с помощью стандартной веб-страницы "Резервное онлайн-копирование" веб-сервера (Страница 1059).

Технические данные

A

A.1 Веб-сайт Siemens для онлайн-поддержки

Техническую информацию об этих продуктах можно найти на веб-сайтеонлайн-поддержки для промышленности Siemens (https://support.industry.siemens.com/cs/us/en/).

A.2 Общие технические данные

Соответствие стандартам

Система автоматизации S7-1200 соответствует следующим нормам/стандартам и техническим условия испытаний. Критерии проверки для системы автоматизации S7-1200 основаны на этих стандартах/нормах и технических условиях испытаний.

Следует учитывать, что возможно не все варианты S7-1200 будут сертифицированы согласно этим стандартам, и состояние сертификации может изменяться без уведомления. Пользователь самостоятельно должен определить применимые сертификации на основе маркировки продуктов. При необходимости можно обратиться в представительство Siemens для получения списка действующих разрешений для отдельных заказных номеров.

Маркировка CE

Система автоматизации S7-1200 удовлетворяет требованиям и целям обеспечения безопасности перечисленных ниже директив ЕС и соответствует согласованным

Европейским стандартам (EN) для программируемых контроллеров, опубликованным в официальных бюллетенях Европейского сообщества.

- Директива EC 2006/95/EC (Директива по низкому напряжению) "Электрическое оборудование, спроектированное для использования в определенных границах напряжения"
  - EN 61131-2 Программируемые контроллеры - Требования к оборудованию и испытания
- Директива EC 2004/108/EC (Директива по ЭМС) "Электромагнитная совместимость"
  - Генерируемые помехи
    EN 61000-6:+A1: Промышленная среда
  - Подавление радиопомех
    EN 61000-6-2: Промышленная среда
- Директива EC 94/9/EG (ATEX) "Оборудование и защитные системы для использования по прямому назначению во взрывоопасных зонах"
  - EN 60079-0:+A11
  - EN 60079-15: вид взрывозащиты 'n':

Декларация соответствия требованиям EC хранится для предоставления всем компетентным органам власти по адресу:

Siemens AG
DI FA AS
Gleiwitzer Str. 555
90475 Nürnberg
Германия

Сертификация по cULus



Underwriters Laboratories, Inc. соответствует:

- Underwriters Laboratories, Inc.: UL 508 Listed (промышленные устройства управления)
- Canadian Standards Association: CSA C22.2 Nummer 142 (оборудование для управления процессами)

Примечание

Серия SIMATIC S7-1200 удовлетворяет стандарту CSA.

Маркировка cULus указывает на то, что S7-1200 был проверен и сертифицирован Underwriters Laboratories (UL) в соответствии со стандартами UL 508 и CSA 22.2 No. 142.

Сертификация по FM



Сертификация по FM
Стандартный класс допуска к эксплуатации 3600 и 3611 (ANSI/UL 121201), 3810 (ANSI/UL 61010-1),
CSA стандарт C22.2 No. 0-10, C22.2 No. 213, C22.2 No. 61010-1
Допущено для использования в:
Class I, Division 2, Gas Group A, B, C, D, Temperature Class T3C Ta = 60 °C [CA, US]

Class I, Zone 2, Group IIC, Temperature Class T3 Ta = 60 °C [US]
Canadian Class I, Zone 2 Installation B CEC 18-150 [CA]

ВАЖНОЕ ИСКЛЮЧЕНИЕ: См. Технические данные для количества входов и выходов, активированных одновременно. Для некоторых моделей норма снижается для Ta = 60 °C.

⚠ **ПРЕДУПРЕЖДЕНИЕ**
Замена компонентов может негативно повлиять на пригодность для Class I, Division 2 и Zone 2. Ремонт устройств должно выполняться только авторизованным сервисным центром Siemens.

Сертификация по IECEx

IEC 60079-0: Взрывоопасные среды – общие требования

IEC 60079-15: Электрический прибор для потенциально взрывоопасных сред;
вид взрывозащиты 'nA'
IEC FMG 14.0012X
Ex nA IIC T3 Gc

IECEx маркировка может появиться на продукте вместе с маркировкой FM для опасных зон.

Разрешены только изделия, имеющие маркировку IECEx. При необходимости можно обратиться в представительство Siemens для получения списка действующих разрешений для отдельных заказных номеров.

Релейные модели не сертифицированы по IECEx.

Использовать маркировку конкретного продукта для оценки температуры.

Монтировать модули в подходящем корпусе, обеспечивающем минимальную степень защиты IP54 согласно IEC 60079-15.

Сертификация по ATEX



ATEX допуск применим только к вариантам для постоянного тока. Сертификация по ATEX не применима к релейным вариантам и вариантам с переменным током.

EN 60079-0: Взрывоопасные среды – общие требования

EN 60079-15: Электрический прибор для потенциально взрывоопасных сред;
вид взрывозащиты 'nA'
II 3 G Ex nA IIC T4 или T3 Gc

Специальные требования для безопасного использования:

Устанавливать модули в подходящем корпусе, обеспечивающем минимальную степень защиты IP54 согласно EN 60529 в месте, обеспечивающем эквивалентную степень защиты.

Присоединенные кабели и проводники должны быть классифицированы для фактической температуры, измеренной в расчетных условиях.

Должны быть приняты меры для исключения превышения для номинального напряжения на клеммах источника питания при переходных помехах более 119 В.

Австралия и Новая Зеландия - RCM Mark (Regulatory Compliance Mark)



Система автоматизации S7-1200 удовлетворяет требованиям стандартов/норм B AS/NZS 61000.6.4 и IEC 610000-6-4 (класс A).

Сертификация в Корее



Система автоматизации S7-1200 удовлетворяет требованиям корейской сертификации (KC маркировка). Она определена как оборудование класса A, предназначена для промышленного применения и не рассматривается для бытового применения.

Разрешение для Евразийского Таможенного Союза (Белоруссия, Казахстан, Российская Федерация)



EAC (Евразийское соответствие): Декларация о соответствии техническому регламенту Таможенного Союза (Technical Regulation of Customs Union, TR CU).

Морские сертификаты

Изделия S7-1200 регулярно предоставляются специальным агентствам для получения одобрений, связанных с определенными рынками и приложениями. При необходимости можно обратиться в представительство Siemens для получения списка действующих разрешений для отдельных заказных номеров.

Классификационные общества:

- ABS (American Bureau of Shipping) США
- Bureau Veritas (BV): Франция
- Det Norske Veritas (DNV): Норвегия
- Germanischer Lloyd (GL): Германия
- Lloyds Register of Shipping (LRS): Англия
- Nippon Kaiji Kyokai (ClassNK): Япония
- Korean Register of Shipping: Корея
- China Classification Society (CSS): Китай

Промышленная среда

Система автоматизации S7-1200 разработана для использования в промышленной среде.

Таблица A-1 Промышленная среда

Область применения	Требования к генерируемым помехам	Требования к подавлению помех
Промышленность	EN 61000-6-4:2007+A1	EN 61000-6-2:2005

Примечание

Система автоматизации S7-1200 предназначена для использования в промышленных зонах; использование в жилом секторе может отрицательно сказаться на теле- или радиоприеме. При использовании S7-1200 в жилом секторе, необходимо обеспечить соответствие уровня генерируемых помех предельному значению класса B по EN 55011.

Примеры соответствующих мер для достижения уровня подавления помех класса B:

- Установка S7-1200 в заземленном шкафу управления

- Использование противопомеховых фильтров в линиях питания

Необходимо подтвердить, что генерируемые помехи соответствуют классу B по EN 55011.

Требуется индивидуальная приемка (завершенная установка должна удовлетворять всем требованиям по безопасности и ЭМС для установки в жилых помещениях).

Электромагнитная совместимость (ЭМС)

Электромагнитная совместимость (ЭМС) - это способность электрического устройства работать по назначению в электромагнитной среде и не излучать электромагнитные помехи, которые могут негативно повлиять на другие электрические устройства поблизости.

Таблица A-2 Стойкость к воздействию помех EN 61000-6-2

Электромагнитная совместимость - подавление помех по EN 61000-6-2	
EN 61000-4-2 Электростатический разряд	8 кВ разряд через воздух на всех поверхностях 6 кВ разряд через контакт с открытыми проводящими поверхностями
EN 61000-4-3 Стойкость к воздействию высокочастотных электромагнитных полей	от 80 до 1000 МГц, 10 В/м, 80 % AM при 1 кГц от 1,4 до 2,0 ГГц, 3 В/м, 80 % AM при 1 кГц от 2,0 до 2,7 ГГц, 1 В/м, 80 % AM при 1 кГц
EN 61000-4-4 Быстрые переходные помехи	2 кВ, 5 кГц в цепи связи с системой питания AC и DC 2 кВ, 5 кГц на клемме входа или выхода
EN 6100-4-5 Устойчивость к динамическим изменениям напряжения питания	Системы AC - 2 кВ синфазная / 1 кВ симметричная помеха Системы DC - 2 кВ синфазная / 1 кВ симметричная помеха Для систем DC, см. "Устойчивость к динамическим изменениям напряжения питания"

Электромагнитная совместимость - подавление помех по EN 61000-6-2	
EN 61000-4-6 Кондуктивные помехи	от 150 кГц до 80 МГц, 10 В эфф., 80% AM при 1 кГц
EN 61000-4-11 Кратковременные посадки напряжения	Системы переменного тока 0% в течение 1 цикла, 40% в течение 12 циклов и 70% в течение 30 циклов при 60 Гц

Таблица A-3 Кондуктивные и излучаемые помехи по EN 61000-6-4

Электромагнитная совместимость - Кондуктивные и излучаемые помехи по EN 61000-6-4		
Кондуктивные помехи EN 55016, класс A, группа 1	от 0,15 МГц до 0,5 МГц	<79 дБ (мкВ) квазипиковое; <66 дБ (мкВ) среднее значение
	от 0,5 МГц до 5 МГц	<73 дБ (мкВ) квазипиковое; <60 дБ (мкВ) среднее значение
	от 5 МГц до 30 МГц	<73 дБ (мкВ) квазипиковое; <60 дБ (мкВ) среднее значение
Излучаемые помехи EN 55016, класс A, группа 1	от 30 МГц до 230 МГц	<40 дБ (мкВ/м) квазипиковое; измерено на расстоянии в 10 м
	от 230 МГц до 1 ГГц	<47 дБ (мкВ/м) квазипиковое; измерено на расстоянии в 10 м
	от 1 ГГц до 3 ГГц	<76 дБ (мкВ/м) квазипиковое; измерено на расстоянии в 10 м

Устойчивость к динамическим изменениям напряжения питания

Системы проводки, подверженные импульсным перегрузкам от удара молнии, должны быть оснащены внешней защитой. Спецификацию для оценки защиты от грозовых перенапряжений можно найти в EN 61000-4-5. Границы устойчивой работы перечислены в EN 61000-6-2. Если S7-1200 CPU и сигнальные модули постоянного тока согласно этому стандарту классифицируются как подверженные импульсным напряжениям, то для обеспечения устойчивой работы потребуется внешняя защита.

Ниже перечислены некоторые устройства, которые обеспечивают необходимую устойчивость к динамическим изменениям напряжения питания. Эти устройства обеспечивают защиту только в том случае, если они правильно установлены в соответствии с рекомендациями изготовителя. Устройства других марок с идентичными или улучшенными техническими данными также могут использоваться:

Таблица A-4 Устройства, обеспечивающие устойчивость к динамическим изменениям напряжения питания

Подсистема	**Устройство защиты**
Электропитание +24 В DC	BLITZDUCTOR VT, BVT AVD 24, инвентарный номер 918 422
Industrial Ethernet	DEHNpatch DPA M CLE RJ45B 48, инвентарный номер 929 121
RS-485	BLITZDUCTOR XT, базовый блок BXT BAS, инвентарный номер 920 300
	BLITZDUCTOR XT, модуль BXT ML2 BD HFS 5, инвентарный номер 920 271
RS-232	BLITZDUCTOR XT, базовый блок BXT BAS, инвентарный номер 920 300
	BLITZDUCTOR XT, модуль BXT ML2 BE S 12, инвентарный номер 920 222

Подсистема	Устройство защиты
Цифровые входы +24 В DC	DEHN, Inc., Type DCO SD2 E 24, инвентарный номер 917 988
Цифровые выходы и питание датчиков +24 В DC	DEHN, Inc., Type DCO SD2 E 24, инвентарный номер 917 988
Аналоговые I/O	DEHN, Inc., Type DCO SD2 E 12, инвентарный номер 917 987
Релейные выходы	Защита не требуется

Условия окружающей среды

Таблица A-5 Транспортировка и хранение

Условия окружающей среды - Транспортировка и хранение	
EN 60068-2-2, тест Bb, сухое тепло и EN 60068-2-1, тест Ab, холод	от -40 °C до +70°C
EN 60068230, тест Db, влажное тепло	от 25 °C до 55 °C, ОВВ 95%
EN 60068-2-14, тест Na, резкий скачок температуры	от -40 °C до +70°C, время выдержки 3 часа, 2 цикла
EN 60068-2-32 свободное падение	0,3 м, 5 раз, в упаковке для транспортировки
Атмосферное давление	от 1140 до 660 гПа (соответствует высоте от -1000 до 3500 м)

Таблица A-6 Климатические условия окружающей среды

Условия окружающей среды - Климатические условия окружающей среды	
Система автоматизации S7-1200 может использоваться в защищенных от непогоды стационарных местах. Условия эксплуатации основаны на требованиях стандарта DIN IEC 60721-3-3: • Класс 3M3 (механические требования) • Класс 3K3 (климатические требования)	
Температура окружающей среды (забор воздуха на 25 мм ниже устройства)	от -20 °C до 60 °C при горизонтальном монтаже от -20 °C до 50 °C при вертикальном монтаже Неконденсируемая влажность 95% Если не определено иначе
Атмосферное давление	от 1140 до 795 гПа (соответствует высоте от -1000 до 2000 м)
Концентрация загрязнений	SO_2: < 0,5 ч./млн.; H_2S: < 0,1 ч./млн.; ОВВ < 60% без конденсации
	ISA-S71.04 уровень сложности G1, G2, G3
EN 60068-2-14, тест Na, изменение температуры	от 0 °C до +60°C
EN 60068227 механическая ударная нагрузка	15 G, импульс 11 мс, 6 ударов по каждой из 3 осей
EN 6006826 синусоидальные вибрационные воздействия	Монтаж на DIN-рейку: 3,5 мм в диапазоне 5-9 Гц, 1 G в диапазоне 8,4 - 150 Гц Установка на панель: 7,0 мм в диапазоне 5-8,4 Гц, 2 G в диапазоне 8,4 - 150 Гц 10 отклонений на ось, 1 октава в минуту

Степень загрязнения и категория перенапряжения по IEC 61131-2

- Степень загрязнения 2
- Категория перенапряжения: II

Вид защиты

- Класс защиты II В EN 61131-2 (защитный проводник не требуется)

Степень защиты

- IP20 механическая защита, EN 60529
- Защита от прямого контакта с высоким напряжением, как например, при тестировании стандартным пробником. Необходима внешняя защита от пыли, грязи, воды и посторонних предметов диаметром < 12,5 мм.

Расчетные напряжения

Таблица A-7 Расчетные напряжения

Номинальное напряжение	Допуск
24 В DC	от 20,4 В DC до 28,8 В DC
120/230 В AC	от 85 В AC до 264 В AC, 47 до 63 Гц

Примечание

Если механический контакт включает выходное напряжение на S7-1200 CPU или цифровой модуль расширения, то в течение приблизительно 50 мкс сигнал "1" подается на цифровые выходы. Это может вызвать непредсказуемое поведение оборудования и, как следствие, привести к тяжким телесным повреждениям и/или материальному ущербу. Следует учитывать эту особенность, в первую очередь при работе с устройствами, реагирующими на короткие импульсы.

Защита от неправильной полярности

Защитой от неправильной полярности оснащается каждая из пар клемм питания +24 В DC или внешнего входного питания для CPU, сигнальных модулей (SM) и сигнальных плат (SB). Однако существует вероятность повреждения системы при соединении различных пар клемм с противоположной полярностью.

Некоторые входные порты 24 В DC системы S7-1200 соединены друг с другом, при этом общий логический нулевой провод соединяет между собой несколько M клемм. Например, следующие цепи тока соединены друг с другом, если в технических данных они обозначены как "не изолированные": источник питания 24 В DC для CPU, питание датчика CPU, вход питания для катушки реле в SM, а также источник питания для

неизолированного аналогового входа. Все не изолированные M клеммы должны быть подсоединены к одному и тому же внешнему опорному потенциалу.

⚠ ПРЕДУПРЕЖДЕНИЕ

Подключение не изолированных M клемм к различным опорным потенциалам вызовет появление непредусмотренных токов, которые могут привести к повреждению или к непредсказуемому поведению целевой системы и подключенного оборудования.

Несоблюдение этих рекомендаций может причинить вред или вызвать непредсказуемое поведение и, как следствие, может привести к тяжким телесным повреждениям и/или материальному ущербу.

Все не изолированные M клеммы в системе S7-1200 всегда должны подключаться к одному и тому же опорному потенциалу.

Выходы постоянного тока

Выходы постоянного тока на CPU, сигнальных модулях (SM) и сигнальных платах (SB) не оснащены схемой защиты цепи от короткого замыкания.

Срок службы реле

Типовые характеристики, сделанные на основании выборочных испытаний, приведены ниже. Фактические данные могут варьироваться в зависимости от приложения. Внешняя цепь защиты, согласованная с нагрузкой, увеличивает срок службы контактов. У NC-контактов типовой ресурс составляет приблизительно одну треть от ресурса NO-контактов при индуктивных или ламповых нагрузках.

Внешняя цепь защиты увеличивает срок службы контактов.

Таблица A-8 Типовые характеристики

Данные для выбора исполнительного элемента	
Непрерывный тепловой ток	Макс. 2 A
Коммутационная способность и ресурс контактов	

Данные для выбора исполнительного элемента				
	Для омической нагрузки	Напряжение	Ток	Кол-во коммутационных циклов (тип.)
		24 В DC	2,0 A	0,1 млн
		24 В DC	1,0 A	0,2 млн
		24 В DC	0,5 A	1,0 млн
		48 В AC	1,5 A	1,5 млн
		60 В AC	1,5 A	1,5 млн
		120 В AC	2,0 A	1,0 млн
		120 В AC	1,0 A	1,5 млн
		120 В AC	0,5 A	2,0 млн
		230 В AC	2,0 A	1,0 млн
		230 В AC	1,0 A	1,5 млн
		230 В AC	0,5 A	2,0 млн
	Для индуктивной нагрузки (согласно IEC 947-5-1 DC13/AC15)	Напряжение	Ток	Кол-во коммутационных циклов (тип.)
		24 В DC	2,0 A	0,05 млн
		24 В DC	1,0 A	0,1 млн
		24 В DC	0,5 A	0,5 млн
		24 В AC	1,5 A	1,0 млн
		48 В AC	1,5 A	1,0 млн
		60 В AC	1,5 A	1,0 млн
		120 В AC	2,0 A	0,7 млн
		120 В AC	1,0 A	1,0 млн
		120 В AC	0,5 A	1,5 млн
		230 В AC	2,0 A	0,7 млн
		230 В AC	1,0 A	1,0 млн
		230 В AC	0,5 A	1,5 млн
Активация цифрового входа		Возможна		
Частота коммутации				
	Механическая	Макс. 10 Гц		
	Для омической нагрузки	Макс. 1 Гц		
	Для индуктивной нагрузки (согласно IEC 947-5-1 DC13/AC15)	Макс. 0,5 Гц		
	Для ламповой нагрузки	Макс. 1 Гц		

Сохранение во внутренней памяти CPU

- Время поддержания сохраняющихся данных и данных архивов: 10 лет
- Сохраняющиеся данные при отключении питания, циклы записи: 2 млн циклов
- Данные архивов данных: циклы записи: 500 млн записей архива данных

Примечание

Влияние архивов данных на внутреннюю память CPU

Каждый процесс записи архива данных использует как минимум 2 кБ памяти. Если программа выполняет частые записи небольшого количества данных, то каждая из записей будет занимать не менее 2 кБ памяти. Лучше собирать небольшие объемы данных в блоках данных (DB), а затем блоки данных записывать в архив данных с менее частыми интервалами.

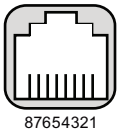
Если программа выполняет множество записей в архив данных, то необходимо рассмотреть вопрос об использовании сменной карты памяти SD.

A.3 Разводка контактов PROFINET интерфейсный порт X1

S7-1200 CPU подключается к сети PROFINET с помощью стандартного гнездового разъема RJ45. Разводка контактов зависит от типа CPU.

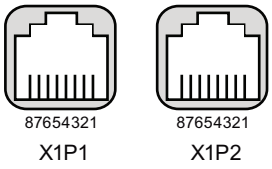
Однопортовые CPU

Однопортовые CPU (CPU 1211C, CPU 1212C и CPU 1214C) имеют следующую стандартную Ethernet-MDI разводку контактов:

Контакт	Имя сигнала	Описание	Разводка контактов гнездового разъема RJ45
1	TD+	Отправка данных	 X1P1
2	TD-		
3	RD+	Прием данных	
4	GND	Масса	
5	GND		
6	RD-	Прием данных	
7	GND	Масса	
8	GND		

Двухпортовые CPU

Двухпортовые CPU (CPU 1215C и CPU1217C) имеют следующую стандартную Ethernet-MDI-X разводку контактов:

Контакт	Имя сигнала	Описание	Разводка контактов гнездового разъема RJ45
1	RD+	Прием данных	
2	RD-		
3	TD+	Отправка данных	
4	GND	Масса	
5	GND		
6	TD-	Отправка данных	
7	GND	Масса	
8	GND		

Примечание

Двухпортовый CPU

Число под контактом указывает на то, что двухпортовые CPU не имеют перекрестных соединений между контактами. В устройствах имеется внутренний коммутатор Ethernet: пары TD +/- и RD +/- не имеют внутренних перекрестных соединений.

Автоопределение

Если конфигурация порта поддерживает автоопределение, S7-1200 CPU автоматически определяет тип кабеля и при необходимости меняет местами линии приема / передачи. Если конфигурация порта отключает автоопределение, CPU также отключает автоматическую перестановку. Параметры автоопределения для порта конфигурируются в диалоговом окне настроек для порта в TIA Portal. Это расширенная опция для порта для интерфейса PROFINET (X1) в свойствах CPU. См. "Конфигурирование порта PROFINET" в разделе 11.2.3.4: "Конфигурирование IP-адреса для CPU в проекте" (Страница 819) для получения дополнительной информации.

A.4 CPU 1211C

A.4.1 Общие технические данные и характеристики

Таблица A-9 Общие технические данные

Технические данные	CPU 1211C AC/DC/RLY	CPU 1211C DC/DC/RLY	CPU 1211C DC/DC/DC
Заказной номер	6ES7211-1BE40-0XB0	6ES7211-1HE40-0XB0	6ES7211-1AE40-0XB0
Габаритные размеры (Ш x В x Г) (мм)	90 x 100 x 75		

Технические данные	CPU 1211C AC/DC/RLY	CPU 1211C DC/DC/RLY	CPU 1211C DC/DC/DC
Вес с упаковкой	420 г	380 г	370 г
Рассеиваемая мощность	10 Вт	8 Вт	
Доступный электрический ток (CM шина)	Макс. 750 мА (5 В DC)		
Доступный электрический ток (24 В DC)	Макс. 300 мА (питание датчиков)		
Потребление тока цифровым входом (24 В DC)	4 мА/вход		

Таблица A-10 Характеристики CPU

Технические данные		Описание
Память пользователя (См. "Общие технические данные" (Страница 1433), "Сохранение во внутренней памяти CPU".)	Рабочая память	50 кБ
	Загружаемая память	1 МБ, внутренняя, расширяемая до объема SD-карты
	С сохранением	10 кБ
Встроенные цифровые I/O		6 входов/4 выхода
Встроенные аналоговые I/O		2 входа
Размер образа процесса		1024 байта для входов (I)/1024 байта для выходов (Q)
Битовая память (M)		4096 байт
Временная (локальная) память		• 16 кбайт для запуска и программного цикла, (включая соответствующие FB и FC) • 6 кбайт для каждого из других уровней приоритета прерываний (включая FB и FC)
Дополнительные сигнальные модули		Нет
Расширение с помощью SB, CB, BB		Макс. 1
Дополнительные коммуникационные модули		Макс. 3 CM
Высокоскоростные счётчики		До 6 сконфигурированных для использования любых встроенных или SB входов. См. "Разводка соединителей аппаратного входа" (Страница 583) для CPU 1211C: Назначение адресов по умолчанию для HSC. 100/[1]80 кГц (Ia.0 до Ia.5)
Импульсные выходы [2]		До 4 сконфигурированных для использования любых встроенных или SB выходов. 100 кГц (Qa.0 до Qa.3)
Входы для захвата импульсов		6
Прерывания по задержке времени		4 с разрешением в 1 мс
Циклические прерывания		4 с разрешением в 1 мс
Прерывания по фронту		6 по нарастающему и 6 по спадающему (10 и 10 с дополнительной сигнальной платой)
Карта памяти		Карта памяти SIMATIC (опция)

Технические данные	Описание
Точность часов реального времени	+/- 60 сек в месяц
Буферизация часов реального времени	Тип. 20 дней/мин. 12 дней при 40 °C (необслуживаемый суперконденсатор)

[1] Более медленная скорость, если высокоскоростной счетчик сконфигурирован для квадратурного A/B режима работы.

[2] Для вариантов CPU с релейными выходами для использования импульсных выходов должна быть установлена цифровая сигнальная плата (SB).

Таблица A-11 Производительность

Тип инструкции		Скорость выполнения	
		Прямая адресация (I, Q и M)	Доступ к DB
Булево значение		0,08 мкс/инструкция	
Перемещение	Move_Bool	0,3 мкс/инструкция	1,17 мкс/инструкция
	Move_Word	0,137 мкс/инструкция	1,0 мкс/инструкция
	Move_Real	0,72 мкс/инструкция	1,0 мкс/инструкция
Математической операции сложения с плавающей запятой	Add Real	1,48 мкс/инструкция	1,78 мкс/инструкция

Примечание

Измеренное время зависит от множества факторов. Приведенные выше показатели производительности относятся к самым быстрым инструкциям в этой категории и к безошибочным программам.

A.4.2 Поддерживаемые CPU 1211C таймеры, счётчики и блоки кода

Таблица A-12 Поддерживаемые CPU 1211C блоки, таймеры и счётчики

Элемент		Описание
Блоки	Тип	OB, FB, FC, DB
	Размер	До размера рабочей памяти
	Количество	В сумме до 1024 блоков (OB + FB + FC + DB)
	Диапазон адресов для FB, FC и DB	FB и FC: от 1 до 65535 (например, от FB 1 до FB 65535) DB: от 1 до 59999
	Глубина вложенности	16 для OB циклов или пусковых OB 6 для любых OB прерывающих событий[1]
	Мониторинг	Контроль состояния 2 блоков кода одновременно.

Элемент		Описание
OB	Цикл программы	Несколько
	Пуск	Несколько
	Прерывание по задержке времени	4 (1 на событие)
	Циклические прерывания	4 (1 на событие)
	Аппаратные прерывания	50 (1 на событие)
	Прерывания по ошибке времени	1
	Диагностические прерывания	1
	Извлечение или вставка модулей	1
	Ошибка стойки или станции	1
	Прерывание по времени	Несколько
	Состояние	1
	Обновление	1
	Профиль	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
	MC-PostServo	1
Таймеры	Тип	IEC
	Количество	Ограничивается только объемом памяти
	Сохранение	Структура в DB, 16 байт на таймер
Счетчики	Тип	IEC
	Количество	Ограничивается только объемом памяти
	Сохранение	Структура в DB, размер зависит от типа счёта • SInt, USInt: 3 байта • Int, UInt 6 байт • DInt, UDInt: 12 байтов

[1] Программы обеспечения безопасности используют два уровня вложенности. Таким образом, программа пользователя имеет в программах обеспечения безопасности четыре уровня вложенности.

Таблица A-13 Коммуникация

Технические данные	Описание
Интерфейсы	1
Тип	Ethernet
Устройство HMI	4
Программатор (PG)	1

Технические данные	Описание
Соединения	• 8 открытых коммуникационных соединений пользователя (активных или пассивных): TSEND_C, TRCV_C, TCON, TDISCON, TSEND и TRCV • 8 CPU/CPU соединений (клиент или сервер) для GET/PUT данных • 6 соединений для динамического распределения на GET/PUT или открытых коммуникационных соединений пользователя
Скорость передачи данных	10/100 Мбит/с
Электрическое разделение (между внешним сигналом и логикой)	Изолирующий трансформатор, 1500 В AC (типовое испытание)[1]
Тип кабеля	CAT5e экранированный
Интерфейсы	
Число интерфейсов PROFINET	1
Число интерфейсов PROFIBUS	0
Интерфейс	
Оборудование для интерфейса	
Число портов	1
Встроенный коммутатор	Нет
RJ-45 (Ethernet)	Да; X1
Протоколы	
PROFINET IO контроллер	Да
PROFINET IO устройство	Да
Коммуникация SIMATIC	Да
Открытый обмен данными через IE	Да
Веб-сервер	Да
Резервирование среды	Нет
PROFINET IO контроллер	
Службы	
Коммуникация PG/OP	Да
S7-маршрутизация	Да
Изохронный режим	Нет
Открытый обмен данными через IE	Да
IRT	Нет
MRP	Нет
PROFIenergy	Да. S7-1200 CPU поддерживает только объект PROFIenergy (с функциональностью I-устройства).
Пуск по приоритету	Да (до 16 устройств PROFINET)
Макс. число подключаемых IO-устройств	16
Макс. число подключаемых IO-устройств для RT	16
Из них в ряд, макс.	16
Макс. число одновременно активируемых/деактивируемых IO-устройств	8

Технические данные		Описание
Время обновления		Минимальное значение для времени обновления также зависит от настроенных для PROFINET IO коммуникационных компонентов, числа IO-устройств и количества сконфигурированных данных пользователя.
C RT		
Такт передачи в 1 мс		От 1 мс до 512 мс
PROFINET IO устройство		
Службы		
Коммуникация PG/OP		Да
S7-маршрутизация		Да
Изохронный режим		Нет
Открытый обмен данными через IE		Да
IRT, поддерживается		Нет
MRP, поддерживается		Нет
PROFIenergy		Да
Устройство общего доступа		Да
Макс. количество IO-контроллеров с устройством общего доступа		2
Коммуникация SIMATIC		
S7-связь, как сервер		Да
S7-связь, как клиент		Да
Макс. данных пользователя за операцию		См. систему интерактивной помощи (S7-связь, объем данных пользователя)
Открытый обмен данными через IE		
TCP/IP:		Да
	Макс. длина данных	8 КБ
	Поддержка для нескольких пассивных соединений на порт	Да
ISO-on-TCP (RFC1006):		Да
	Макс. длина данных	8 КБ
UDP		Да
	Макс. длина данных	1472 байта
DHCP		Нет
SNMP		Да
DCP		Да
LLDP		Да

[1] Изоляция порта Ethernet предназначена для снижение рисков из-за опасных напряжений при кратковременных сбоях сети. Она не соответствует требованиям безопасности для обычных развязок по напряжению в сетях переменного тока.

Таблица A-14 Электропитание

Технические данные		**CPU 1211C AC/DC/RLY**	**CPU 1211C DC/DC/RLY**	**CPU 1211C DC/DC/DC**
Диапазон напряжений		от 85 до 264 В AC	от 20,4 В DC до 28,8 В DC	
Частота сети		от 47 до 63 Гц	--	
Входной ток	только CPU при макс. нагрузке	60 мА при 120 В AC 30 мА при 240 В AC	300 мА при 24 В DC	300 мА при 24 В DC
	CPU со всеми модулями расширения при макс. нагрузке	180 мА при 120 В AC 90 мА при 240 В AC	900 мА при 24 В DC	
Пусковой ток (макс.)		20 А при 264 В AC	12 А при 28,8 В DC	
$I^2 t$		0,8 A^2 с	0,5 A^2 с	
Электрическое разделение (между внешним питанием и логикой)		1500 В AC	Без электрического разделения	
Ток утечки на землю, от линии переменного тока на функциональную землю		макс. 0,5 мА	--	
Допустимый перерыв в питании		20 мА при 120 В AC 80 мА при 240 В AC	10 мс при 24 В DC	
Встроенный предохранитель, недоступен для пользователя		3 А, 250 В, плавкий предохранитель		

Таблица A-15 Электропитание датчиков

Технические данные	**CPU 1211C AC/DC/RLY**	**CPU 1211C DC/DC/RLY**	**CPU 1211C DC/DC/DC**
Диапазон напряжений	от 20,4 до 28,8 В DC	L+ - 4 В DC (мин.)	
Номинальный выходной ток (макс.)	300 мА (с защитой от коротких замыканий)		
Пульсации напряжения (до 10 МГц), не более	< 1 В между пиками	Как в цепи питания	
Электрическая развязка (между логикой CPU и электропитанием датчиков)	Без электрического разделения		

A.4.3 Цифровые входы и выходы

Таблица A-16 Цифровые входы

Технические данные	**CPU 1211C AC/DC/RLY, CPU 1211C DC/DC/RLY и CPU 1211C DC/DC/DC**
Количество входов	6
Тип	Со снижением тока/с увеличением тока (IEC Typ 1, если со снижением тока)
Номинальное напряжение	24 В DC при 4 мА, ном.
Длительно допустимое входное напряжение	До 30 В DC
Перенапряжение	35 В DC на 0,5 с

Технические данные	**CPU 1211C AC/DC/RLY, CPU 1211C DC/DC/RLY и CPU 1211C DC/DC/DC**
Сигнал логический 1 (мин.)	15 В DC при 2,5 мА
Сигнал логический 0 (макс.)	5 В DC при 1 мА
Электрическое разделение (между полевой стороной и логикой)	707 В DC (типовое испытание)
Изолированные группы	1
Время фильтрации	мкс установка: 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0 мс установка: 0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0
Скорость изменения входного сигнала для высокоскоростных счетчиков (макс.) (логический уровень 1 = от 15 до 26 В DC)	100/80 кГц (от Ia.0 до Ia.5)
Количество одновременно активных входов	6 при 60 °C для горизонтальной установки, 50 °C для вертикальной установки
Длина кабеля (в метрах)	500 м экранированный, 300 м неэкранированный, 50 м экранированный для входов высокоскоростных счетчиков

Таблица A-17 Цифровые выходы

Технические данные	**CPU 1211C AC/DC/RLY и CPU 1211C DC/DC/RLY**	**CPU 1211C DC/DC/DC**
Выходы	4	
Тип	Замыкающие контакты реле	Транзисторные ключи MOSFET
Диапазон напряжений	от 5 до 30 В DC или от 5 до 250 В AC	от 20,4 до 28,8 В DC
Сигнал логический 1 при макс. токе	--	Мин. 20 В DC
Сигнал логический 0 при нагрузке 10 кОм	--	Макс. 0,1 В DC
Ток (макс.)	2,0 А	0,5 А
Ламповая нагрузка	30 Вт DC/200 Вт AC	5 Вт
Сопротивление в активном состоянии	Макс. 0,2 Ом для нового модуля	Макс. 0,6 Ом
Ток утечки на выход	--	До 10 мкА
Импульсный ток выхода, не более	7 А через замкнутый контакт	8 А макс. в течение 100 мс
Защита от перегрузки	Нет	
Электрическое разделение (между полевой стороной и логикой)	1500 В AC (катушка - контакт) Нет (катушка - логика)	707 В DC (типовое испытание)
Изолированные группы	1	
Индуктивное фиксирующее напряжение	--	L+ - 48 В DC, 1 Вт рассеиваемая мощность
Макс. частота коммутации реле	1 Гц	--
Задержка переключения (от Qa.0 до Qa.3)	До 10 мс	До 1,0 мкс из Выкл на Вкл До 3,0 мкс из Вкл на Выкл
Частота следования импульсов	Не рекомендуется [1]	100 кГц (Qa.0 до Qa.3)[2], мин. 2 Гц
Механический срок службы (без нагрузки)	10.000.000 циклов переключения	--

Технические данные	CPU 1211C AC/DC/RLY и CPU 1211C DC/DC/RLY	CPU 1211C DC/DC/DC
Срок службы контактов при номинальной нагрузке	100.000 циклов переключения	--
Поведение при переходе из RUN в STOP	Последнее значение или замещающее значение (по умолчанию 0)	
Количество одновременно активных выходов	4 при 60 °C для горизонтальной установки, 50 °C для вертикальной установки	
Длина кабеля (в метрах)	500 м экранированный, 150 м неэкранированный	

[1] Для вариантов CPU с релейными выходами при использования импульсных выходов должна быть установлена цифровая сигнальная плата (SB).

[2] В зависимости от используемого приемника импульсов и кабеля, дополнительный нагрузочный резистор (по крайней мере 10% от номинального тока) может улучшить качество импульсных сигналов и стойкость к воздействию помех.

A.4.4 Аналоговые входы

Таблица A-18 Аналоговые входы

Технические данные	Описание
Количество входов	2
Тип	Напряжение (асимметричный)
Диапазон изменения сигналов	от 0 до 10 В
Цифровое представление полной шкалы (слово данных)	от 0 до 27648
Выход за границы диапазона	от 10,001 до 11,759 В
Выход за границы диапазона (слово данных)	от 27649 до 32511
Переполнение	от 11,760 до 11,852 В
Переполнение (слово данных)	от 32512 до 32767
Разрешение	10 бит
Максимальное входное напряжение	35 В DC
Сглаживание	Отсутствует, слабое, среднее или сильное См. таблицу Реакция на скачок (мс) для аналоговых входов CPU (Страница 1453).
Подавление помех	10, 50 или 60 Гц
Полное сопротивление	≥100 кОм
Электрическое разделение (между полевой стороной и логикой)	Нет
Точность (25 °C / -20 до 60 °C)	3,0% / 3,5% по отношению к конечной точке шкалы
Длина кабеля (в метрах)	100 м, экранированная витая пара

A.4.4.1 Реакция на скачок встроенных аналоговых входов CPU

Таблица A-19 Реакция на скачок (мс), от 0 до 10 В измеренная при 95%

Выбор сглаживания (усреднение по выборке)	**Частота подавления помех (время интегрирования)**		
	60 Гц	**50 Гц**	**10 Гц**
Отсутствует (1 цикл): без усреднения	50 мс	50 мс	100 мс
Слабое (4 цикла): 4 значения	60 мс	70 мс	200 мс
Среднее (16 циклов) 16 значений	200 мс	240 мс	1150 мс
Сильное (32 цикла) 32 значения	400 мс	480 мс	2300 мс
Время выборки	**4,17 мс**	**5 мс**	**25 мс**

A.4.4.2 Время выборки для встроенных аналоговых портов CPU

Таблица A-20 Время выборки встроенных аналоговых входов CPU

Частота подавления помех (выбор времени интегрирования)	**Время выборки**
60 Гц (16,6 мс)	4,17 мс
50 Гц (20 мс)	5 мс
10 Гц (100 мс)	25 мс

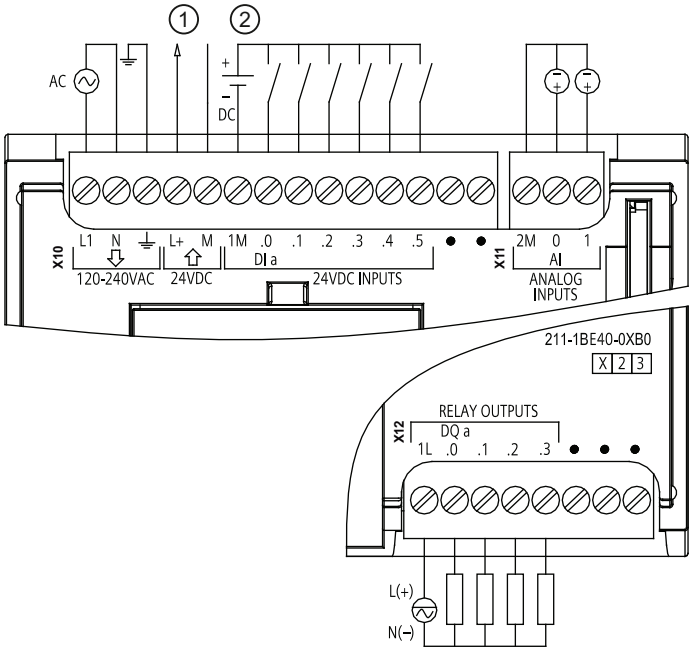
A.4.4.3 Диапазоны измерений для аналоговых входов напряжения (CPU)

Таблица A-21 Представление аналогового входа для напряжения (CPU)

Система		**Диапазон измерения напряжения**	
Десятичная	**Шестнадцатеричная**	**от 0 до 10 В**	
32767	7FFF	11,852 В	Выход за верхний установленный предел
32512	7F00		
32511	7EFF	11,759 В	Выход за границы диапазона
27649	6C01		
27648	6C00	10 В	Расчетный диапазон
20736	5100	7,5 В	
34	22	12 мВ	
0	0	0 В	
Отрицательные значения		Отрицательные значения не поддерживаются	

A.4.5 Схемы электрических соединений CPU 1211C

Таблица A-22 CPU 1211C AC/DC/RLY (6ES7211-1BE40-0XB0)



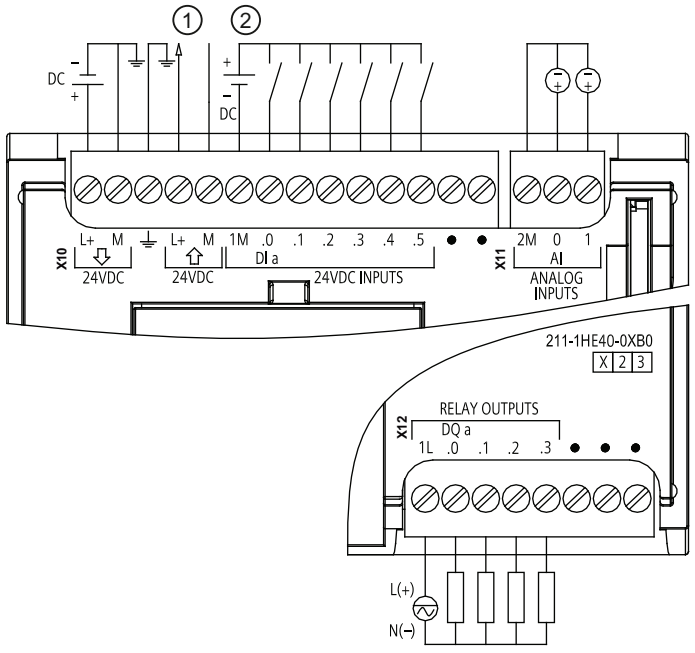
①	Выход питания датчиков 24 В DC Для дополнительной помехоустойчивости соединить "M" с корпусом, даже если питание датчика не используется.
②	Для входов-потребителей соединить "-" с "M" (см. рисунок). Для входов-источников соединить "+" с "M".
Примечание 1: Соединители X11 должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	
Примечание 2: Одна из двух клемм L1 или N (L2) может быть подключена к источнику напряжения до 240 В AC. Клемма N может рассматриваться как L2 и ее не нужно заземлять. Для клемм L1 и N (L2) не требуется защита от неправильной полярности.	
Примечание 3: Дополнительную информацию о Ethernet порте CPU см. в Конфигурация устройства (Страница 151).	

Таблица A-23 Разводка контактов для CPU 1211C AC/DC/RLY (6ES7211-1BE40-0XB0)

Контакт	X10	X11 (позолоченный)	X12
1	L1 / 120-240 В AC	2 M	1L
2	N / 120-240 В AC	AI 0	DQ a.0
3	Функциональное заземление	AI 1	DQ a.1
4	L+ / питание датчиков 24 В DC	--	DQ a.2
5	M / питание датчиков 24 В DC	--	DQ a.3
6	1M	--	Не подключено

Контакт	X10	X11 (позолоченный)	X12
7	DI a.0	--	Не подключено
8	DI a.1	--	Не подключено
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	Не подключено	--	--
14	Не подключено	--	--

Таблица A-24 CPU 1211C DC/DC/RLY (6ES7211-1HE40-0XB0)



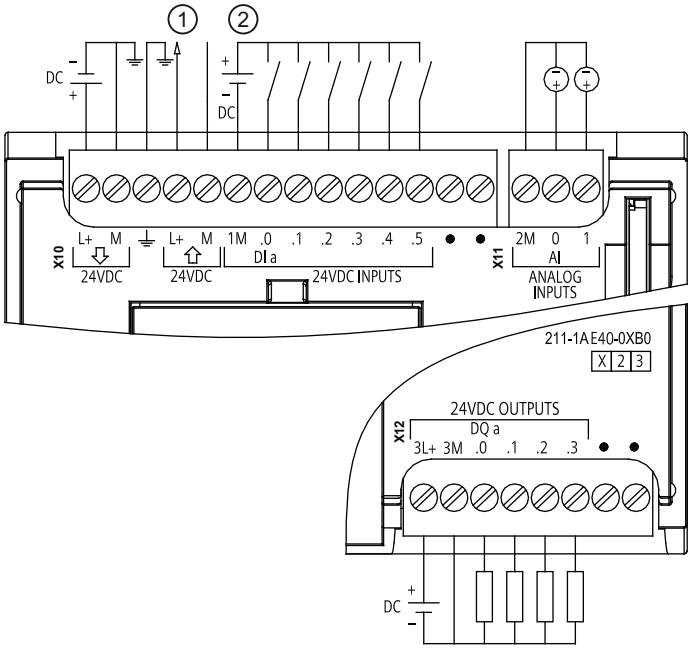
①	Питание датчиков 24 В DC Для дополнительной помехоустойчивости соединить "M" с корпусом, даже если питание датчика не используется.
②	Для входов-потребителей соединить "-" с "M" (см. рисунок). Для входов-источников соединить "+" с "M".
Примечание 1: Соединители X11 должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	
Примечание 2: Дополнительную информацию о Ethernet порте CPU см. в Конфигурация устройства (Страница 151).	

Таблица A-25 Разводка контактов для CPU 1211C DC/DC/RLY (6ES7211-1HE40-0XB0)

Контакт	X10	X11 (позолоченный)	X12
1	L+ / 24 В DC	2 M	1L
2	M / 24 В DC	AI 0	DQ a.0
3	Функциональное заземление	AI 1	DQ a.1
4	L+ / питание датчиков 24 В DC	--	DQ a.2
5	M / питание датчиков 24 В DC	--	DQ a.3

Контакт	X10	X11 (позолоченный)	X12
6	1M	--	Не подключено
7	DI a.0	--	Не подключено
8	DI a.1	--	Не подключено
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	Не подключено	--	--
14	Не подключено	--	--

Таблица A-26 CPU 1211C DC/DC/DC (6ES7211-1AE40-0XB0)



①	Питание датчиков 24 В DC Для дополнительной помехоустойчивости соединить "M" с корпусом, даже если питание датчика не используется.
②	Для входов-потребителей соединить "-" с "M" (см. рисунок). Для входов-источников соединить "+" с "M".
Примечание 1: Соединители X11 должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	
Примечание 2: Дополнительную информацию о Ethernet порте CPU см. в Конфигурация устройства (Страница 151).	

Таблица A-27 Разводка контактов для CPU 1211C DC/DC/DC (6ES7211-1AE40-0XB0)

Контакт	X10	X11 (позолоченный)	X12
1	L+ / 24 В DC	2 M	3L+
2	M / 24 В DC	AI 0	3M
3	Функциональное заземление	AI 1	DQ a.0
4	L+ / питание датчиков 24 В DC	--	DQ a.1

Контакт	X10	X11 (позолоченный)	X12
5	M / питание датчиков 24 В DC	--	DQ a.2
6	1M	--	DQ a.3
7	DI a.0	--	Не подключено
8	DI a.1	--	Не подключено
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	Не подключено	--	--
14	Не подключено	--	--

Примечание

Неиспользуемые аналоговые входы должны быть закорочены.

A.5 CPU 1212C

A.5.1 Общие технические данные и характеристики

Таблица A-28 Общее

Технические данные	CPU 1212C AC/DC/RLY	CPU 1212C DC/DC/RLY	CPU 1212C DC/DC/DC
Заказной номер	6ES7212-1BE40-0XB0	6ES7212-1HE40-0XB0	6ES7212-1AE40-0XB0
Габаритные размеры (Ш x В x Г) (мм)	90 x 100 x 75		
Вес с упаковкой	425 г	385 г	370 г
Рассеиваемая мощность	11 Вт	9 ВТ	
Доступный электрический ток (SM и CM шина)	Макс. 1000 мА (5 В DC)		
Доступный электрический ток (24 В DC)	Макс. 300 мА (питание датчиков)		
Потребление тока цифровым входом (24 В DC)	4 мА/вход		

Таблица A-29 Характеристики CPU

Технические данные		**Описание**
Память пользователя (См. "Общие технические данные" (Страница 1433), "Сохранение во внутренней памяти CPU".)	Рабочая память	75 кБ
	Загружаемая память	2 МБ, внутренняя, расширяемая до объема SD-карты
	С сохранением	10 кБ
Встроенные цифровые I/O		8 входов/6 выходов
Встроенные аналоговые I/O		2 входа
Размер образа процесса		1024 байта для входов (I)/1024 байта для выходов (Q)
Битовая память (M)		4096 байт
Временная (локальная) память		• 16 кбайт для запуска и программного цикла, (включая соответствующие FB и FC) • 6 кбайт для каждого из других уровней приоритета прерываний (включая FB и FC)
Дополнительные сигнальные модули		Макс. 2 SM
Расширение с помощью SB, CB, BB		Макс. 1
Дополнительные коммуникационные модули		Макс. 3 CM
Высокоскоростные счётчики		До 6 сконфигурированных для использования любых встроенных или SB входов. См. "Разводка соединителей аппаратного входа" (Страница 583) для CPU 1212C: Назначение адресов по умолчанию для HSC. • 100/[1]80 кГц (Ia.0 до Ia.5) • 30/[1]20 кГц (Ia.6 до Ia.7)
Импульсные выходы [2]		До 4 сконфигурированных для использования любых встроенных или SB выходов. • 100 кГц (Qa.0 до Qa.3) • 20 кГц (Qa.4 до Qa.5)
Входы для захвата импульсов		8
Прерывания по задержке времени		4 всего с разрешением в 1 мс
Циклические прерывания		4 всего с разрешением в 1 мс
Прерывания по фронту		8 по нарастающему и 8 по спадающему (12 и 12 с дополнительной сигнальной платой)
Карта памяти		Карта памяти SIMATIC (опция)
Точность часов реального времени		+/- 60 сек в месяц
Буферизация часов реального времени		Тип. 20 дней/мин. 12 дней при 40 °C (необслуживаемый суперконденсатор)

[1] Более медленная скорость, если высокоскоростной счетчик сконфигурирован для квадратурного A/B режима работы.

[2] Для вариантов CPU с релейными выходами для использования импульсных выходов должна быть установлена цифровая сигнальная плата (SB).

Таблица A-30 Производительность

Тип инструкции		**Скорость выполнения**	
		Прямая адресация (I, Q и M)	**Доступ к DB**
Булево значение		0,08 мкс/инструкция	
Перемещение	Move_Bool	0,3 мкс/инструкция	1,17 мкс/инструкция
	Move_Word	0,137 мкс/инструкция	1,0 мкс/инструкция
	Move_Real	0,72 мкс/инструкция	1,0 мкс/инструкция
Математической операции сложения с плавающей запятой	Add Real	1,48 мкс/инструкция	1,78 мкс/инструкция

Примечание

Измеренное время зависит от множества факторов. Приведенные выше показатели производительности относятся к самым быстрым инструкциям в этой категории и к безошибочным программам.

A.5.2 Поддерживаемые CPU 1212C таймеры, счётчики и блоки кода

Таблица A-31 Поддерживаемые CPU 1212C блоки, таймеры и счётчики

Элемент		**Описание**
Блоки	Тип	OB, FB, FC, DB
	Размер	До размера рабочей памяти
	Количество	В сумме до 1024 блоков (OB + FB + FC + DB)
	Диапазон адресов для FB, FC и DB	FB и FC: от 1 до 65535 (например, от FB 1 до FB 65535) DB: от 1 до 59999
	Глубина вложенности	16 для OB циклов или пусковых OB 6 для любых OB прерывающих событий[1]
	Мониторинг	Контроль состояния 2 блоков кода одновременно.

Элемент		Описание
OB	Цикл программы	Несколько
	Пуск	Несколько
	Прерывание по задержке времени	4 (1 на событие)
	Циклические прерывания	4 (1 на событие)
	Аппаратные прерывания	50 (1 на событие)
	Прерывания по ошибке времени	1
	Диагностические прерывания	1
	Извлечение или вставка модулей	1
	Ошибка стойки или станции	1
	Прерывание по времени	Несколько
	Состояние	1
	Обновление	1
	Профиль	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
	MC-PostServo	1
Таймеры	Тип	IEC
	Количество	Ограничивается только объемом памяти
	Сохранение	Структура в DB, 16 байт на таймер
Счетчики	Тип	IEC
	Количество	Ограничивается только объемом памяти
	Сохранение	Структура в DB, размер зависит от типа счёта • SInt, USInt: 3 байта • Int, UInt 6 байт • DInt, UDInt: 12 байтов

[1] Программы обеспечения безопасности используют два уровня вложенности. Таким образом, программа пользователя имеет в программах обеспечения безопасности четыре уровня вложенности.

Таблица A-32 Коммуникация

Технические данные	Описание
Интерфейсы	1
Тип	Ethernet
Устройство HMI	4
Программатор (PG)	1

Технические данные	Описание
Соединения	• 8 открытых коммуникационных соединений пользователя (активных или пассивных): TSEND_C, TRCV_C, TCON, TDISCON, TSEND и TRCV • 8 CPU/CPU соединений (клиент или сервер) для GET/PUT данных • 6 соединений для динамического распределения на GET/PUT или открытых коммуникационных соединений пользователя
Скорость передачи данных	10/100 Мбит/с
Электрическое разделение (между внешним сигналом и логикой)	Изолирующий трансформатор, 1500 В AC (типовое испытание)[1]
Тип кабеля	CAT5e экранированный
Интерфейсы	
Число интерфейсов PROFINET	1
Число интерфейсов PROFIBUS	0
Интерфейс	
Оборудование для интерфейса	
Число портов	1
Встроенный коммутатор	Нет
RJ-45 (Ethernet)	Да; X1
Протоколы	
PROFINET IO контроллер	Да
PROFINET IO устройство	Да
Коммуникация SIMATIC	Да
Открытый обмен данными через IE	Да
Веб-сервер	Да
Резервирование среды	Нет
PROFINET IO контроллер	
Службы	
Коммуникация PG/OP	Да
S7-маршрутизация	Да
Изохронный режим	Нет
Открытый обмен данными через IE	Да
IRT	Нет
MRP	Нет
PROFIenergy	Да. S7-1200 CPU поддерживает только объект PROFIenergy (с функциональностью I-устройства).
Пуск по приоритету	Да (до 16 устройств PROFINET)
Макс. число подключаемых IO-устройств	16
Макс. число подключаемых IO-устройств для RT	16
Из них в ряд, макс.	16
Макс. число одновременно активируемых/деактивируемых IO-устройств	8

Технические данные		Описание
Время обновления		Минимальное значение для времени обновления также зависит от настроенных для PROFINET IO коммуникационных компонентов, числа IO-устройств и количества сконфигурированных данных пользователя.
C RT		
Такт передачи в 1 мс		От 1 мс до 512 мс
PROFINET IO устройство		
Службы		
Коммуникация PG/OP		Да
S7-маршрутизация		Да
Изохронный режим		Нет
Открытый обмен данными через IE		Да
IRT, поддерживается		Нет
MRP, поддерживается		Нет
PROFIenergy		Да
Устройство общего доступа		Да
Макс. количество IO-контроллеров с устройством общего доступа		2
Коммуникация SIMATIC		
S7-связь, как сервер		Да
S7-связь, как клиент		Да
Макс. данных пользователя за операцию		См. систему интерактивной помощи (S7-связь, объем данных пользователя)
Открытый обмен данными через IE		
TCP/IP:		Да
	Макс. длина данных	8 КБ
	Поддержка для нескольких пассивных соединений на порт	Да
ISO-on-TCP (RFC1006):		Да
	Макс. длина данных	8 КБ
UDP		Да
	Макс. длина данных	1472 байта
DHCP		Нет
SNMP		Да
DCP		Да
LLDP		Да

[1] Изоляция порта Ethernet предназначена для снижение рисков из-за опасных напряжений при кратковременных сбоях сети. Она не соответствует требованиям безопасности для обычных развязок по напряжению в сетях переменного тока.

Таблица A-33 Электропитание

Технические данные		**CPU 1212C AC/DC/RLY**	**CPU 1212C DC/DC/RLY**	**CPU 1212C DC/DC/DC**
Диапазон напряжений		от 85 до 264 В AC	от 20,4 В DC до 28,8 В DC	
Частота сети		от 47 до 63 Гц	--	
Пусковой ток (макс. нагрузка)	Только CPU	80 мА при 120 В AC 40 мА при 240 В AC	400 мА при 24 В DC	
	CPU со всеми модулями расширения	240 мА при 120 В AC 120 мА при 240 В AC	1200 мА при 24 В DC	
Пусковой ток (макс.)		20 А при 264 В AC	12 А при 28,8 В DC	
I^2 t		0,8 A^2 с	0,5 A^2 с	
Электрическое разделение (между внешним питанием и логикой)		1500 В AC	Без электрического разделения	
Ток утечки на землю, от линии переменного тока на функциональную землю		макс. 0,5 мА	--	
Допустимый перерыв в питании		20 мА при 120 В AC 80 мА при 240 В AC	10 мс при 24 В DC	
Встроенный предохранитель, недоступен для пользователя		3 А, 250 В, плавкий предохранитель		

Таблица A-34 Электропитание датчиков

Технические данные	**CPU 1212C AC/DC/RLY**	**CPU 1212C DC/DC/RLY**	**CPU 1212C DC/DC/DC**
Диапазон напряжений	от 20,4 до 28,8 В DC	L+ - 4 В Мин. DC	
Номинальный выходной ток (макс.)	300 мА (с защитой от коротких замыканий)		
Пульсации напряжения (до 10 МГц), не более	< 1 В между пиками	Как в цепи питания	
Электрическая развязка (между логикой CPU и электропитанием датчиков)	Без электрического разделения		

A.5.3 Цифровые входы и выходы

Таблица A-35 Цифровые входы

Технические данные	**CPU 1212C AC/DC/RLY, DC/DC/RLY и DC/DC/DC**
Количество входов	8
Тип	Потребитель/источник (IEC Тур 1, если потребитель)
Номинальное напряжение	24 В DC при 4 мА, ном.
Длительно допустимое входное напряжение	До 30 В DC
Перенапряжение	35 В DC на 0,5 с
Сигнал логический 1 (мин.)	15 В DC при 2,5 мА

Технические данные	**CPU 1212C AC/DC/RLY, DC/DC/RLY и DC/DC/DC**
Сигнал логический 0 (макс.)	5 В DC при 1 мА
Электрическое разделение (между полевой стороной и логикой)	707 В DC (типовое испытание)
Изолированные группы	1
Время фильтрации	мкс установка: 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0 мс установка: 0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0
Скорость изменения входного сигнала для высокоскоростных счетчиков (макс.) (логический уровень 1 = от 15 до 26 В DC)	100/80 кГц (от Ia.0 до Ia.5) 30/20 кГц (Ia.6 до Ia.7)
Количество одновременно активных входов	4 (без смежных точек) при 60 °C для горизонтальной установки или 50 °C для вертикальной установки 8 при 55 °C для горизонтальной установки или 45 °C для вертикальной установки
Длина кабеля (в метрах)	500 м экранированный, 300 м неэкранированный, 50 м экранированный для входов высокоскоростных счетчиков

Таблица A-36 Цифровые выходы

Технические данные	**CPU 1212C AC/DC/RLY и DC/DC/RLY**	**CPU 1212C DC/DC/DC**
Выходы	6	
Тип	Замыкающие контакты реле	Транзисторные ключи MOSFET
Диапазон напряжений	от 5 до 30 В DC или от 5 до 250 В AC	от 20,4 до 28,8 В DC
Сигнал логический 1 при макс. токе	--	Мин. 20 В DC
Сигнал логический 0 при нагрузке 10 кОм	--	Макс. 0,1 В DC
Ток (макс.)	2,0 А	0,5 А
Ламповая нагрузка	30 Вт DC/200 Вт AC	5 Вт
Сопротивление в активном состоянии	Макс. 0,2 Ом для нового модуля	Макс. 0,6 Ом
Ток утечки на выход	--	До 10 мкА
Импульсный ток выхода, не более	7 А через замкнутый контакт	8 А макс. в течение 100 мс
Защита от перегрузки	Нет	
Электрическое разделение (между полевой стороной и логикой)	1500 В AC (катушка - контакт) Нет (катушка - логика)	707 В DC (типовое испытание)
Изолированные группы	2	1
Электрическая развязка (между группами)	1500 В AC [1]	--
Индуктивное фиксирующее напряжение	--	L+ - 48 В DC, 1 Вт рассеиваемая мощность
Задержка переключения (от Qa.0 до Qa.3)	До 10 мс	До 1,0 мкс из Выкл на Вкл До 3,0 мкс из Вкл на Выкл
Задержка переключения (от Qa.4 до Qa.5)	До 10 мс	До 5 мкс из Выкл на Вкл До 20 мкс из Вкл на Выкл
Макс. частота коммутации реле	1 Гц	--

Технические данные	CPU 1212C AC/DC/RLY и DC/DC/RLY	CPU 1212C DC/DC/DC
Частота следования импульсов	Не рекомендуется[2]	100 кГц (от Qa.0 до Qa.3)[3], 2 Гц мин. 20 кГц (от Qa.4 до Qa.5)[3]
Механический срок службы (без нагрузки)	10.000.000 циклов переключения	--
Срок службы контактов при номинальной нагрузке	100.000 циклов переключения	--
Поведение при переходе из RUN в STOP	Последнее значение или замещающее значение (по умолчанию 0)	Последнее значение или замещающее значение (по умолчанию 0)
Количество одновременно активных выходов	3 (без смежных точек) при 60 °C для горизонтальной установки или 50 °C для вертикальной установки 6 при 55 °C для горизонтальной установки или 45 °C для вертикальной установки	
Длина кабеля (в метрах)	500 м экранированный, 150 м неэкранированный	

1 Электрическое разделение между группами RLY развязывает сетевое напряжение и SELV/PELV, а также различные фазы до 250 В AC и линию заземления.

2 Для вариантов CPU с релейными выходами для использования импульсных выходов должна быть установлена цифровая сигнальная плата (SB).

3 В зависимости от используемого приемника импульсов и кабеля, дополнительный нагрузочный резистор (по крайней мере 10% от номинального тока) может улучшить качество импульсных сигналов и стойкость к воздействию помех.

A.5.4 Аналоговые входы

Таблица A-37 Аналоговые входы

Технические данные	Описание
Количество входов	2
Тип	Напряжение (асимметричный)
Диапазон изменения сигналов	от 0 до 10 В
Цифровое представление полной шкалы (слово данных)	от 0 до 27648
Выход за границы диапазона	от 10,001 до 11,759 В
Выход за границы диапазона (слово данных)	от 27649 до 32511
Переполнение	от 11,760 до 11,852 В
Переполнение (слово данных)	от 32512 до 32767
Разрешение	10 бит
Максимальное входное напряжение	35 В DC
Сглаживание	Отсутствует, слабое, среднее или сильное См. таблицу Реакция на скачок (мс) для аналоговых входов CPU (Страница 1466).
Подавление помех	10, 50 или 60 Гц
Полное сопротивление	≥100 кОм

Технические данные	Описание
Электрическое разделение (между полевой стороной и логикой)	Нет
Точность (25 °C / -20 до 60 °C)	3,0% / 3,5% по отношению к конечной точке шкалы
Длина кабеля (в метрах)	100 м, экранированная витая пара

A.5.4.1 Реакция на скачок встроенных аналоговых входов CPU

Таблица A-38 Реакция на скачок (мс), от 0 до 10 В измеренная при 95%

Выбор сглаживания (усреднение по выборке)	Частота подавления помех (время интегрирования)		
	60 Гц	50 Гц	10 Гц
Отсутствует (1 цикл): без усреднения	50 мс	50 мс	100 мс
Слабое (4 цикла): 4 значения	60 мс	70 мс	200 мс
Среднее (16 циклов) 16 значений	200 мс	240 мс	1150 мс
Сильное (32 цикла) 32 значения	400 мс	480 мс	2300 мс
Время выборки	**4,17 мс**	**5 мс**	**25 мс**

A.5.4.2 Время выборки для встроенных аналоговых портов CPU

Таблица A-39 Время выборки встроенных аналоговых входов CPU

Частота подавления помех (выбор времени интегрирования)	Время выборки
60 Гц (16,6 мс)	4,17 мс
50 Гц (20 мс)	5 мс
10 Гц (100 мс)	25 мс

A.5.4.3 Диапазоны измерений для аналоговых входов напряжения (CPU)

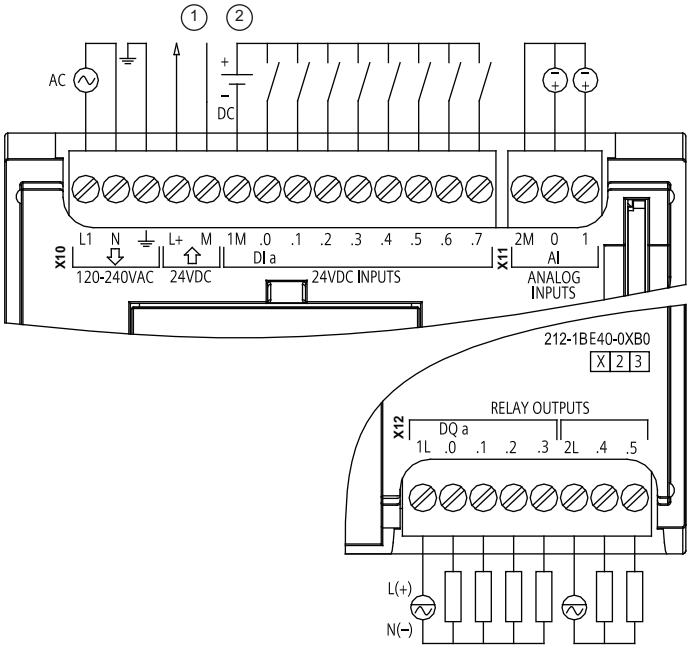
Таблица A-40 Представление аналогового входа для напряжения (CPU)

Система		Диапазон измерения напряжения	
Десятичная	Шестнадцатеричная	от 0 до 10 В	
32767	7FFF	11,852 В	Выход за верхний установленный предел
32512	7F00		
32511	7EFF	11,759 В	Выход за границы диапазона
27649	6C01		

Система		Диапазон измерения напряжения	
Десятичная	**Шестнадцате-ричная**	**от 0 до 10 В**	
27648	6C00	10 В	Расчетный диапазон
20736	5100	7,5 В	
34	22	12 мВ	
0	0	0 В	
Отрицательные значения		Отрицательные значения не поддерживаются	

A.5.5 Схемы электрических соединений CPU 1212C

Таблица A-41 CPU 1212C AC/DC/RLY (6ES7212-1BE40-0XB0)

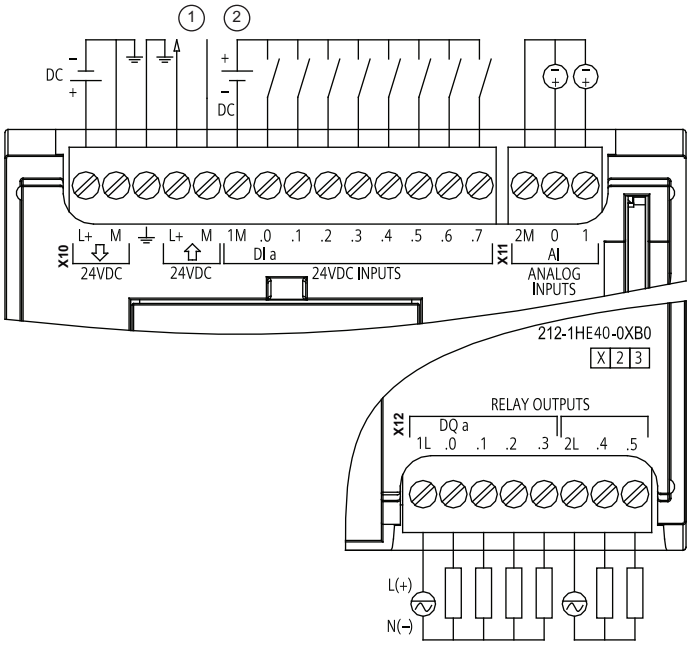


①	Питание датчиков 24 В DC Для дополнительной помехоустойчивости соединить "M" с корпусом, даже если питание датчика не используется.
②	Для входов-потребителей соединить "-" с "M" (см. рисунок). Для входов-источников соединить "+" с "M".
Примечание 1: Соединители X11 должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	
Примечание 2: Одна из двух клемм L1 или N (L2) может быть подключена к источнику напряжения до 240 В AC. Клемма N может рассматриваться как L2 и ее не нужно заземлять. Для клемм L1 и N (L2) не требуется защита от неправильной полярности.	
Примечание 3: Дополнительную информацию о Ethernet порте CPU см. в Конфигурация устройства (Страница 151).	

Таблица A-42 Разводка контактов для CPU 1212C AC/DC/RLY (6ES7212-1BE40-0XB0)

Контакт	X10	X11 (позолоченный)	X12
1	L1 / 120-240 В AC	2 M	1L
2	N / 120-240 В AC	AI 0	DQ a.0
3	Функциональное заземление	AI 1	DQ a.1
4	L+ / питание датчиков 24 В DC	--	DQ a.2
5	M / питание датчиков 24 В DC	--	DQ a.3
6	1M	--	2L
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

Таблица A-43 CPU 1212C DC/DC/RLY (6ES7212-1HE40-0XB0)

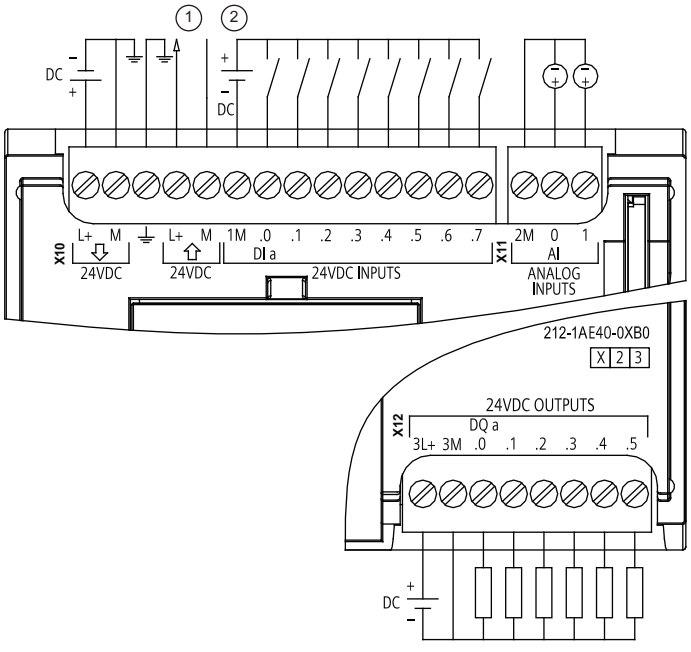


①	Питание датчиков 24 В DC Для дополнительной помехоустойчивости соединить "M" с корпусом, даже если питание датчика не используется.
②	Для входов-потребителей соединить "-" с "M" (см. рисунок). Для входов-источников соединить "+" с "M".
Примечание 1: Соединители X11 должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	
Примечание 2: Дополнительную информацию о Ethernet порте CPU см. в Конфигурация устройства (Страница 151).	

Таблица A-44 Разводка контактов для CPU 1212C DC/DC/RLY (6ES7212-1HE40-0XB0)

Контакт	X10	X11 (позолоченный)	X12
1	L+ / 24 В DC	2 M	1L
2	M / 24 В DC	AI 0	DQ a.0
3	Функциональное заземление	AI 1	DQ a.1
4	L+ / питание датчиков 24 В DC	--	DQ a.2
5	M / питание датчиков 24 В DC	--	DQ a.3
6	1M	--	2L
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

Таблица A-45 CPU 1212C DC/DC/DC (6ES7212-1AE40-0XB0)



①	Питание датчиков 24 В DC Для дополнительной помехоустойчивости соединить "M" с корпусом, даже если питание датчика не используется.
②	Для входов-потребителей соединить "-" с "M" (см. рисунок). Для входов-источников соединить "+" с "M".
Примечание 1: Соединители X11 должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	
Примечание 2: Дополнительную информацию о Ethernet порте CPU см. в Конфигурация устройства (Страница 151).	

Таблица A-46 Разводка контактов для CPU 1212C DC/DC/DC (6ES7212-1AE40-0XB0)

Контакт	X10	X11 (позолоченный)	X12
1	L+ / 24 В DC	2 M	3L+
2	M / 24 В DC	AI 0	3M
3	Функциональное зазе-мление	AI 1	DQ a.0
4	L+ / питание датчиков 24 В DC	--	DQ a.1
5	M / питание датчиков 24 В DC	--	DQ a.2
6	1M	--	DQ a.3
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

Примечание

Неиспользуемые аналоговые входы должны быть закорочены.

A.6 CPU 1214C

A.6.1 Общие технические данные и характеристики

Таблица A-47 Общее

Технические данные	CPU 1214C AC/DC/RLY	CPU 1214C DC/DC/RLY	CPU 1214C DC/DC/DC
Заказной номер	6ES7214-1BG40-0XB0	6ES7214-1HG40-0XB0	6ES7214-1AG40-0XB0
Габаритные размеры (Ш x В x Г) (мм)	110 x 100 x 75		
Вес с упаковкой	475 г	435 г	415 г
Рассеиваемая мощность	14 Вт	12 Вт	
Доступный электрический ток (SM и CM шина)	Макс. 1.600 мА (5 В DC)		

Технические данные	**CPU 1214C AC/DC/RLY**	**CPU 1214C DC/DC/RLY**	**CPU 1214C DC/DC/DC**
Доступный электрический ток (24 В DC)	Макс. 400 мА (питание датчиков)		
Потребление тока цифровым входом (24 В DC)	4 мА/вход		

Таблица A-48 Характеристики CPU

Технические данные		**Описание**
Память пользователя (См. "Общие технические данные" (Страница 1433), "Сохранение во внутренней памяти CPU".)	Рабочая память	100 кБ
	Загружаемая память	4 МБ, внутренняя, расширяемая до объема SD-карты
	С сохранением	10 кБ
Встроенные цифровые I/O		14 входов/10 выходов
Встроенные аналоговые I/O		2 входа
Размер образа процесса		1024 байта для входов (I)/1024 байта для выходов (Q)
Битовая память (M)		8192 байта
Временная (локальная) память		• 16 кбайт для запуска и программного цикла, (включая соответствующие FB и FC) • 6 кбайт для каждого из других уровней приоритета прерываний (включая FB и FC)
Дополнительные сигнальные модули		Макс. 8 SM
Расширение с помощью SB, CB, BB		Макс. 1
Дополнительные коммуникационные модули		Макс. 3 CM
Высокоскоростные счётчики		До 6 сконфигурированных для использования любых встроенных или SB входов. См. "Разводка соединителей аппаратного входа" (Страница 583) для CPU 1214C: Назначение адресов по умолчанию для HSC. • 100/[1]80 кГц (Ia.0 до Ia.5) • 30/[1]20 кГц (Ia.6 до Ib.5)
Импульсные выходы [2]		До 4 сконфигурированных для использования любых встроенных или SB выходов. • 100 кГц (Qa.0 до Qa.3) • 20 кГц (Qa.4 до Qb.1)
Входы для захвата импульсов		14
Прерывания по задержке времени		4 с разрешением в 1 мс
Циклические прерывания		4 с разрешением в 1 мс
Прерывания по фронту		12 по нарастающему и 12 по спадающему (16 и 16 с дополнительной сигнальной платой)
Карта памяти		Карта памяти SIMATIC (опция)

Технические данные	Описание
Точность часов реального времени	+/- 60 сек в месяц
Буферизация часов реального времени	Тип. 20 дней/мин. 12 дней при 40 °C (необслуживаемый суперконденсатор)

[1] Более медленная скорость, если высокоскоростной счетчик сконфигурирован для квадратурного A/B режима работы.

[2] Для вариантов CPU с релейными выходами для использования импульсных выходов должна быть установлена цифровая сигнальная плата (SB).

Таблица A-49 Производительность

Тип инструкции		Скорость выполнения	
		Прямая адресация (I, Q и M)	Доступ к DB
Булево значение		0,08 мкс/инструкция	
Перемещение	Move_Bool	0,3 мкс/инструкция	1,17 мкс/инструкция
	Move_Word	0,137 мкс/инструкция	1,0 мкс/инструкция
	Move_Real	0,72 мкс/инструкция	1,0 мкс/инструкция
Математической операции сложения с плавающей запятой	Add Real	1,48 мкс/инструкция	1,78 мкс/инструкция

Примечание

Измеренное время зависит от множества факторов. Приведенные выше показатели производительности относятся к самым быстрым инструкциям в этой категории и к безошибочным программам.

A.6.2 Поддерживаемые CPU 1214C таймеры, счётчики и блоки кода

Таблица A-50 Поддерживаемые CPU 1214C блоки, таймеры и счётчики

Элемент		Описание
Блоки	Тип	OB, FB, FC, DB
	Размер	OB, FB, FC: 64 кБ DB: до размера рабочей памяти
	Количество	В сумме до 1024 блоков (OB + FB + FC + DB)
	Диапазон адресов для FB, FC и DB	FB и FC: от 1 до 65535 (например, от FB 1 до FB 65535) DB: от 1 до 59999
	Глубина вложенности	16 для OB циклов или пусковых OB 6 для любых OB прерывающих событий[1]
	Мониторинг	Контроль состояния 2 блоков кода одновременно.

Элемент		**Описание**
OB	Цикл программы	Несколько
	Пуск	Несколько
	Прерывания по задержке времени	4 (1 на событие)
	Циклические прерывания	4 (1 на событие)
	Аппаратные прерывания	50 (1 на событие)
	Прерывания по ошибке времени	1
	Диагностические прерывания	1
	Извлечение или вставка модулей	1
	Ошибка стойки или станции	1
	Прерывание по времени	Несколько
	Состояние	1
	Обновление	1
	Профиль	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
	MC-PostServo	1
Таймеры	Тип	IEC
	Количество	Ограничивается только объемом памяти
	Сохранение	Структура в DB, 16 байт на таймер
Счетчики	Тип	IEC
	Количество	Ограничивается только объемом памяти
	Сохранение	Структура в DB, размер зависит от типа счёта • SInt, USInt: 3 байта • Int, UInt: 6 байт • DInt, UDInt: 12 байт

[1] Программы обеспечения безопасности используют два уровня вложенности. Таким образом, программа пользователя имеет в программах обеспечения безопасности четыре уровня вложенности.

Таблица A-51 Коммуникация

Технические данные	**Описание**
Интерфейсы	1
Тип	Ethernet
Устройство HMI	4
Программатор (PG)	1

Технические данные	Описание
Соединения	• 8 открытых коммуникационных соединений пользователя (активных или пассивных): TSEND_C, TRCV_C, TCON, TDISCON, TSEND и TRCV • 8 CPU/CPU соединений (клиент или сервер) для GET/PUT данных • 6 соединений для динамического распределения на GET/PUT или открытых коммуникационных соединений пользователя
Скорость передачи данных	10/100 Мбит/с
Электрическое разделение (между внешним сигналом и логикой)	Изолирующий трансформатор, 1500 В AC (типовое испытание)[1]
Тип кабеля	CAT5e экранированный
Интерфейсы	
Число интерфейсов PROFINET	1
Число интерфейсов PROFIBUS	0
Интерфейс	
Оборудование для интерфейса	
Число портов	1
Встроенный коммутатор	Нет
RJ-45 (Ethernet)	Да; X1
Протоколы	
PROFINET IO контроллер	Да
PROFINET IO устройство	Да
Коммуникация SIMATIC	Да
Открытый обмен данными через IE	Да
Веб-сервер	Да
Резервирование среды	Нет
PROFINET IO контроллер	
Службы	
Коммуникация PG/OP	Да
S7-маршрутизация	Да
Изохронный режим	Нет
Открытый обмен данными через IE	Да
IRT	Нет
MRP	Нет
PROFIenergy	Да. S7-1200 CPU поддерживает только объект PROFIenergy (с функциональностью I-устройства).
Пуск по приоритету	Да (до 16 устройств PROFINET)
Макс. число подключаемых IO-устройств	16
Макс. число подключаемых IO-устройств для RT	16
Из них в ряд, макс.	16
Макс. число одновременно активируемых/деактивируемых IO-устройств	8

Технические данные		Описание
Время обновления		Минимальное значение для времени обновления также зависит от настроенных для PROFINET IO коммуникационных компонентов, числа IO-устройств и количества сконфигурированных данных пользователя.
C RT		
Такт передачи в 1 мс		От 1 мс до 512 мс
PROFINET IO устройство		
Службы		
Коммуникация PG/OP		Да
S7-маршрутизация		Да
Изохронный режим		Нет
Открытый обмен данными через IE		Да
IRT, поддерживается		Нет
MRP, поддерживается		Нет
PROFIenergy		Да
Устройство общего доступа		Да
Макс. количество IO-контроллеров с устройством общего доступа		2
Коммуникация SIMATIC		
S7-связь, как сервер		Да
S7-связь, как клиент		Да
Макс. данных пользователя за операцию		См. систему интерактивной помощи (S7-связь, объем данных пользователя)
Открытый обмен данными через IE		
TCP/IP:		Да
	Макс. длина данных	8 КБ
	Поддержка для нескольких пассивных соединений на порт	Да
ISO-on-TCP (RFC1006):		Да
	Макс. длина данных	8 КБ
UDP		Да
	Макс. длина данных	1472 байта
DHCP		Нет
SNMP		Да
DCP		Да
LLDP		Да

[1] Изоляция порта Ethernet предназначена для снижение рисков из-за опасных напряжений при кратковременных сбоях сети. Она не соответствует требованиям безопасности для обычных развязок по напряжению в сетях переменного тока.

Таблица A-52 Электропитание

Технические данные		**CPU 1214C AC/DC/RLY**	**CPU 1214C DC/DC/RLY**	**CPU 1214C DC/DC/DC**
Диапазон напряжений		от 85 до 264 В AC	от 20,4 В DC до 28,8 В DC	
Частота сети		47 ... 63 Гц	--	
Пусковой ток (макс. нагрузка)	Только CPU	100 мА при 120 В AC 50 мА при 240 В AC	500 мА при 24 В DC	
	CPU со всеми модулями расширения	300 мА при 120 В AC 150 мА при 240 В AC	1500 мА при 24 В DC	
Пусковой ток (макс.)		20 А при 264 В AC	12 А при 28,8 В DC	
I^2t		0,8 A^2 с	0,5 A^2 с	
Электрическое разделение (между внешним питанием и логикой)		1500 В AC	Без электрического разделения	
Ток утечки на землю, от линии переменного тока на функциональную землю		макс. 0,5 мА	-	
Допустимый перерыв в питании		20 мА при 120 В AC 80 мА при 240 В AC	10 мс при 24 В DC	
Встроенный предохранитель, недоступен для пользователя		3 А, 250 В, плавкий предохранитель		

Таблица A-53 Электропитание датчиков

Технические данные	**CPU 1214C AC/DC/RLY**	**CPU 1214C DC/DC/RLY**	**CPU 1214C DC/DC/DC**
Диапазон напряжений	от 20,4 до 28,8 В DC	L+ - 4 В DC (мин.)	
Номинальный выходной ток (макс.)	400 мА (с защитой от коротких замыканий)		
Пульсации напряжения (до 10 МГц), не более	< 1 В между пиками	Как в цепи питания	
Электрическая развязка (между логикой CPU и электропитанием датчиков)	Без электрического разделения		

A.6.3 Цифровые входы и выходы

Таблица A-54 Цифровые входы

Технические данные	**CPU 1214C AC/DC/RLY**	**CPU 1214C DC/DC/RLY**	**CPU 1214C DC/DC/DC**
Количество входов	14		
Тип	Потребитель/источник (IEC Тур 1, если потребитель)		
Номинальное напряжение	24 В DC при 4 мА, ном.		
Длительно допустимое напряжение	До 30 В DC		
Перенапряжение	35 В DC на 0,5 с		
Сигнал логический 1 (мин.)	15 В DC при 2,5 мА		
Сигнал логический 0 (макс.)	5 В DC при 1 мА		

Технические данные	CPU 1214C AC/DC/RLY	CPU 1214C DC/DC/RLY	CPU 1214C DC/DC/DC
Электрическое разделение (между полевой стороной и логикой)	707 В DC (типовое испытание)		
Изолированные группы	1		
Время фильтрации	мкс установка: 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0 мс установка: 0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0		
Скорость изменения входного сигнала для высокоскоростных счетчиков (макс.) (логический уровень 1 = от 15 до 26 В DC)	100/80 кГц (от Ia0 до Ia.5) 30/20 кГц (от Ia6 до Ib.5)		
Количество одновременно активных входов	• 7 (без смежных точек) при 60 °C для горизонтальной установки или 50 °C для вертикальной установки • 14 при 55 °C для горизонтальной установки или 45 °C для вертикальной установки		
Длина кабеля (в метрах)	500 м экранированный, 300 м неэкранированный, 50 м экранированный для входов высокоскоростных счетчиков		

Таблица A-55 Цифровые выходы

Технические данные	CPU 1214C AC/DC/RLY и DC/DC/RLY	CPU 1214C DC/DC/DC
Выходы	10	
Тип	Замыкающие контакты реле	Транзисторные ключи MOSFET
Диапазон напряжений	от 5 до 30 В DC или от 5 до 250 В AC	от 20,4 до 28,8 В DC
Сигнал логический 1 при макс. токе	--	Мин. 20 В DC
Сигнал логический 0 при нагрузке 10 кОм	--	Макс. 0,1 В DC
Ток (макс.)	2,0 А	0,5 А
Ламповая нагрузка	30 Вт DC/200 Вт AC	5 Вт
Сопротивление в активном состоянии	Макс. 0,2 Ом для нового модуля	Макс. 0,6 Ом
Ток утечки на выход	--	До 10 мкА
Импульсный ток выхода, не более	7 А через замкнутый контакт	8 А макс. в течение 100 мс
Защита от перегрузки	Нет	
Электрическое разделение (между полевой стороной и логикой)	1500 В AC (катушка - контакт) Нет (катушка - логика)	707 В DC (типовое испытание)
Изолированные группы	2	1
Электрическая развязка (между группами)	1500 В AC [1]	--
Индуктивное фиксирующее напряжение	--	L+ - 48 В DC, 1 Вт рассеиваемая мощность
Задержка переключения (от Qa.0 до Qa.3)	До 10 мс	До 1,0 мкс из Выкл на Вкл До 3,0 мкс из Вкл на Выкл
Задержка переключения (от Qa.4 до Qb.1)	До 10 мс	До 5 мкс из Выкл на Вкл До 20 мкс из Вкл на Выкл

Технические данные	CPU 1214C AC/DC/RLY и DC/DC/RLY	CPU 1214C DC/DC/DC
Макс. частота коммутации реле	1 Гц	--
Частота следования импульсов	Не рекомендуется [2]	100 кГц (от Qa.0 до Qa.3)[3], 2 Гц мин. 20 кГц (от Qa.4 до Qb.1)[3]
Механический срок службы (без нагрузки)	10.000.000 циклов переключения	--
Срок службы контактов при номинальной нагрузке	100.000 циклов переключения	--
Поведение при переходе из RUN в STOP	Последнее значение или замещающее значение (по умолчанию 0)	
Количество одновременно активных выходов	• 5 (без смежных точек) при 60 °C для горизонтальной установки или 50 °C для вертикальной установки • 10 при 55 °C для горизонтальной установки или 45 °C для вертикальной установки	
Длина кабеля (в метрах)	500 м экранированный, 150 м неэкранированный	

[1] Электрическое разделение между группами RLY развязывает сетевое напряжение и SELV/PELV, а также различные фазы до 250 В AC и линию заземления.

[2] Для вариантов CPU с релейными выходами для использования импульсных выходов должна быть установлена цифровая сигнальная плата (SB).

[3] В зависимости от используемого приемника импульсов и кабеля, дополнительный нагрузочный резистор (по крайней мере 10% от номинального тока) может улучшить качество импульсных сигналов и стойкость к воздействию помех.

A.6.4 Аналоговые входы

Таблица A-56 Аналоговые входы

Технические данные	Описание
Количество входов	2
Тип	Напряжение (асимметричный)
Диапазон изменения сигналов	от 0 до 10 В
Цифровое представление полной шкалы (слово данных)	от 0 до 27648
Выход за границы диапазона	от 10,001 до 11,759 В
Выход за границы диапазона (слово данных)	от 27649 до 32511
Переполнение	от 11,760 до 11,852 В
Переполнение (слово данных)	от 32512 до 32767
Разрешение	10 бит
Максимальное входное напряжение	35 В DC
Сглаживание	Отсутствует, слабое, среднее или сильное См. таблицу Реакция на скачок (мс) для аналоговых входов CPU (Страница 1479).
Подавление помех	10, 50 или 60 Гц
Полное сопротивление	≥100 кОм

Технические данные	Описание
Электрическое разделение (между полевой стороной и логикой)	Нет
Точность (25 °C / -20 до 60 °C)	3,0% / 3,5% по отношению к конечной точке шкалы
Длина кабеля (в метрах)	100 м, экранированная витая пара

A.6.4.1 Реакция на скачок встроенных аналоговых входов CPU

Таблица A-57 Реакция на скачок (мс), от 0 до 10 В измеренная при 95%

Выбор сглаживания (усреднение по выборке)	Частота подавления помех (время интегрирования)		
	60 Гц	50 Гц	10 Гц
Отсутствует (1 цикл): без усреднения	50 мс	50 мс	100 мс
Слабое (4 цикла): 4 значения	60 мс	70 мс	200 мс
Среднее (16 циклов) 16 значений	200 мс	240 мс	1150 мс
Сильное (32 цикла) 32 значения	400 мс	480 мс	2300 мс
Время выборки	**4,17 мс**	**5 мс**	**25 мс**

A.6.4.2 Время выборки для встроенных аналоговых портов CPU

Таблица A-58 Время выборки встроенных аналоговых входов CPU

Частота подавления помех (выбор времени интегрирования)	Время выборки
60 Гц (16,6 мс)	4,17 мс
50 Гц (20 мс)	5 мс
10 Гц (100 мс)	25 мс

A.6.4.3 Диапазоны измерений для аналоговых входов напряжения (CPU)

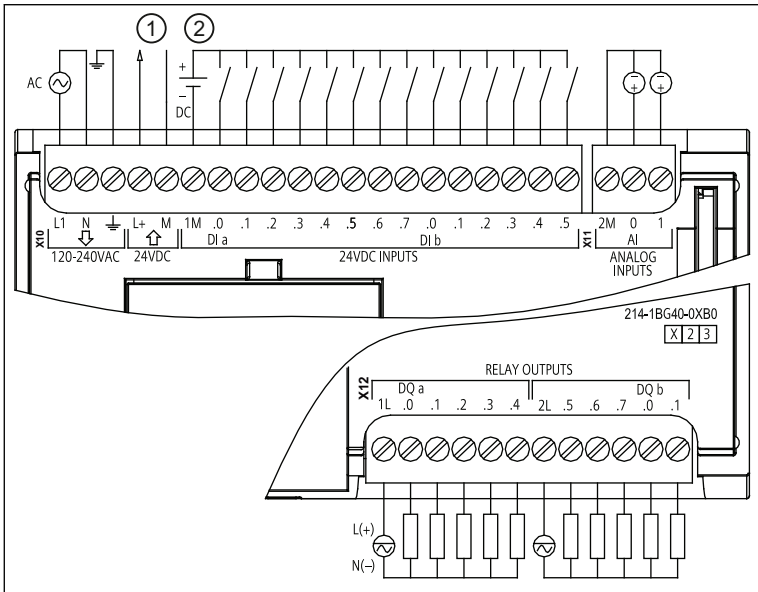
Таблица A-59 Представление аналогового входа для напряжения (CPU)

Система		Диапазон измерения напряжения	
Десятичная	Шестнадцатеричная	от 0 до 10 В	
32767	7FFF	11,852 В	Выход за верхний установленный предел
32512	7F00		
32511	7EFF	11,759 В	Выход за границы диапазона
27649	6C01		

Система		**Диапазон измерения напряжения**	
Десятичная	**Шестнадцате-ричная**	**от 0 до 10 В**	
27648	6C00	10 В	Расчетный диапазон
20736	5100	7,5 В	
34	22	12 мВ	
0	0	0 В	
Отрицательные значения		Отрицательные значения не поддерживаются	

A.6.5 Схемы электрических соединений CPU 1214C

Таблица A-60 CPU 1214C AC/DC/RLY (6ES7214-1BG40-0XB0)

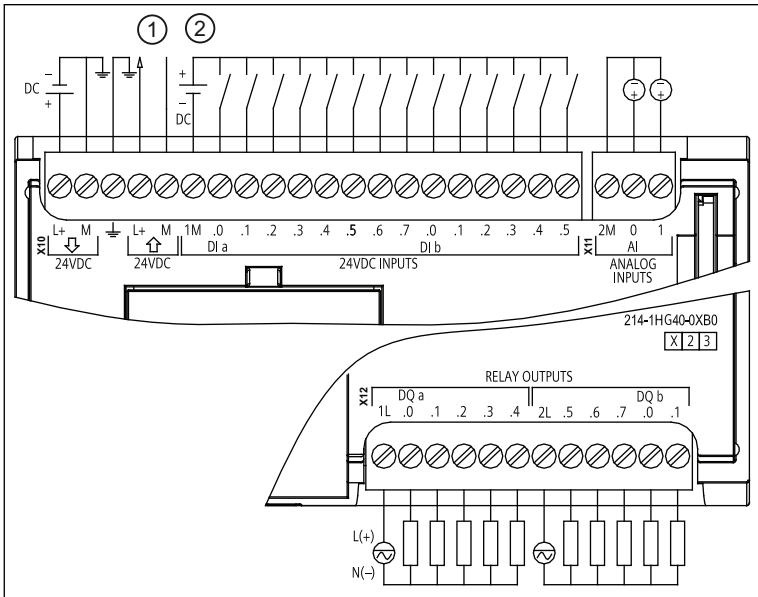


①	Питание датчиков 24 В DC Для дополнительной помехоустойчивости соединить "M" с корпусом, даже если питание датчика не используется.
②	Для входов-потребителей соединить "-" с "M" (см. рисунок). Для входов-источников соединить "+" с "M".
Примечание 1: Соединители X11 должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	
Примечание 2: Одна из двух клемм L1 или N (L2) может быть подключена к источнику напряжения до 240 В AC. Клемма N может рассматриваться как L2 и ее не нужно заземлять. Для клемм L1 и N (L2) не требуется защита от неправильной полярности.	
Примечание 3: Дополнительную информацию о Ethernet порте CPU см. в Конфигурация устройства (Страница 151).	

Таблица A-61 Разводка контактов для CPU 1214C AC/DC/RLY (6ES7214-1BG40-0XB0)

Контакт	X10	X11 (позолоченный)	X12
1	L1 / 120-240 В AC	2 M	1L
2	N / 120-240 В AC	AI 0	DQ a.0
3	Функциональное зазе-мление	AI 1	DQ a.1
4	L+ / питание датчиков 24 В DC	--	DQ a.2
5	M / питание датчиков 24 В DC	--	DQ a.3
6	1M	--	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Таблица A-62 CPU 1214C DC/DC/RLY (6ES7214-1HG40-0XB0)



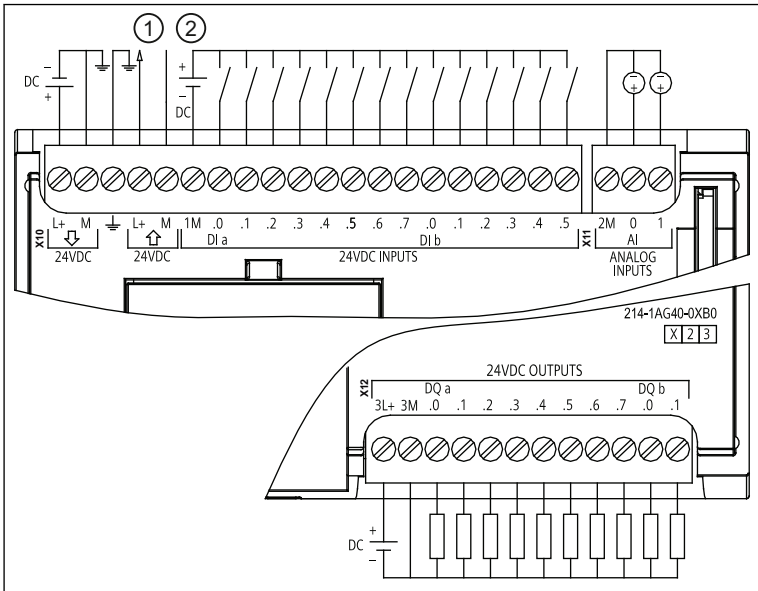
①	Питание датчиков 24 В DC Для дополнительной помехоустойчивости соединить "M" с корпусом, даже если питание датчика не используется.
②	Для входов-потребителей соединить "-" с "M" (см. рисунок). Для входов-источников соединить "+" с "M".
Примечание 1: Соединители X11 должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	
Примечание 2: Дополнительную информацию о Ethernet порте CPU см. в Конфигурация устройства (Страница 151).	

Таблица A-63 Разводка контактов для CPU 1214C DC/DC/RLY (6ES7214-1HG40-0XB0)

Контакт	X10	X11 (позолоченный)	X12
1	L+ / 24 В DC	2 M	1L
2	M / 24 В DC	AI 0	DQ a.0
3	Функциональное заземление	AI 1	DQ a.1
4	L+ / питание датчиков 24 В DC	--	DQ a.2
5	M / питание датчиков 24 В DC	--	DQ a.3
6	1M	--	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--

Контакт	X10	X11 (позолоченный)	X12
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Таблица A-64 CPU 1214C DC/DC/DC (6ES7214-1AG40-0XB0)



1	Питание датчиков 24 В DC Для дополнительной помехоустойчивости соединить "M" с корпусом, даже если питание датчика не используется.
2	Для входов-потребителей соединить "-" с "M" (см. рисунок). Для входов-источников соединить "+" с "M".
Примечание 1: Соединители X11 должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	
Примечание 2: Дополнительную информацию о Ethernet порте CPU см. в Конфигурация устройства (Страница 151).	

Таблица A-65 Разводка контактов для CPU 1214C DC/DC/DC (6ES7214-1AG40-0XB0)

Контакт	X10	X11 (позолоченный)	X12
1	L+ / 24 В DC	2 M	3L+
2	M / 24 В DC	AI 0	3M
3	Функциональное заземление	AI 1	DQ a.0
4	L+ / питание датчиков 24 В DC	--	DQ a.1
5	M / питание датчиков 24 В DC	--	DQ a.2
6	1M	--	DQ a.3
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6

Контакт	X10	X11 (позолоченный)	X12
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	-
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Примечание

Неиспользуемые аналоговые входы должны быть закорочены.

A.7 CPU 1215C

A.7.1 Общие технические данные и характеристики

Таблица A-66 Общее

Технические данные	CPU 1215C AC/DC/RLY	CPU 1215C DC/DC/RLY	CPU 1215C DC/DC/DC
Заказной номер	6ES7215-1BG40-0XB0	6ES7215-1HG40-0XB0	6ES7215-1AG40-0XB0
Габаритные размеры (Ш x В x Г) (мм)	130 x 100 x 75		
Вес с упаковкой	585 г	550 г	520 г
Рассеиваемая мощность	14 Вт	12 Вт	
Доступный электрический ток (SM и CM шина)	Макс. 1600 мА (5 В DC)		
Доступный электрический ток (24 В DC)	Макс. 400 мА (питание датчиков)		
Потребление тока цифровым входом (24 В DC)	4 мА/вход		

Таблица A-67 Характеристики CPU

Технические данные		**Описание**
Память пользователя (См. "Общие технические данные" (Страница 1433), "Сохранение во внутренней памяти CPU".)	Рабочая память	125 кБ
	Загружаемая память	4 МБ, внутренняя, расширяемая до объема SD-карты
	С сохранением	10 кБ
Встроенные цифровые I/O		14 входов/10 выходов
Встроенные аналоговые I/O		2 входа/2 выхода
Размер образа процесса		1024 байта для входов (I)/1024 байта для выходов (Q)
Битовая память (M)		8192 байта
Временная (локальная) память		• 16 кбайт для запуска и программного цикла, (включая соответствующие FB и FC) • 6 кбайт для каждого из других уровней приоритета прерываний (включая FB и FC)
Дополнительные сигнальные модули		Макс. 8 SM
Расширение с помощью SB, CB, BB		Макс. 1
Дополнительные коммуникационные модули		Макс. 3 CM
Высокоскоростные счётчики		До 6 сконфигурированных для использования любых встроенных или SB входов. См. "Разводка соединителей аппаратного входа" (Страница 583) для CPU 1215C: Назначение адресов по умолчанию для HSC. • 100/[1]80 кГц (Ia.0 до Ia.5) • 30/[1]20 кГц (Ia.6 до Ib.5)
Импульсные выходы [2]		До 4 сконфигурированных для использования любых встроенных или SB выходов. • 100 кГц (Qa.0 до Qa.3) • 20 кГц (Qa.4 до Qb.1)
Входы для захвата импульсов		14
Прерывания по задержке времени		4 с разрешением в 1 мс
Циклические прерывания		4 с разрешением в 1 мс
Прерывания по фронту		12 по нарастающему и 12 по спадающему (16 и 16 с дополнительной сигнальной платой)
Карта памяти		Карта памяти SIMATIC (опция)
Точность часов реального времени		+/- 60 сек в месяц
Буферизация часов реального времени		Тип. 20 дней/мин. 12 дней при 40 °C (необслуживаемый суперконденсатор)

[1] Более медленная скорость, если высокоскоростной счетчик сконфигурирован для квадратурного A/B режима работы.

[2] Для вариантов CPU с релейными выходами для использования импульсных выходов должна быть установлена цифровая сигнальная плата (SB).

Таблица A-68 Производительность

Тип инструкции		**Скорость выполнения**	
		Прямая адресация (I, Q и M)	**Доступ к DB**
Булево значение		0,08 мкс/инструкция	
Перемещение	Move_Bool	0,3 мкс/инструкция	1,17 мкс/инструкция
	Move_Word	0,137 мкс/инструкция	1,0 мкс/инструкция
	Move_Real	0,72 мкс/инструкция	1,0 мкс/инструкция
Математической операции сложения с плавающей запятой	Add Real	1,48 мкс/инструкция	1,78 мкс/инструкция

Примечание

Измеренное время зависит от множества факторов. Приведенные выше показатели производительности относятся к самым быстрым инструкциям в этой категории и к безошибочным программам.

A.7.2 Поддерживаемые CPU 1215C таймеры, счётчики и блоки кода

Таблица A-69 Поддерживаемые CPU 1215C блоки, таймеры и счётчики

Элемент		**Описание**
Блоки	Тип	OB, FB, FC, DB
	Размер	OB, FB, FC: 64 кбайт DB: до размера рабочей памяти
	Количество	В сумме до 1024 блоков (OB + FB + FC + DB)
	Диапазон адресов для FB, FC и DB	FB и FC: от 1 до 65535 (например, от FB 1 до FB 65535) DB: от 1 до 59999
	Глубина вложенности	16 для OB циклов или пусковых OB 6 для любых OB прерывающих событий[1]
	Мониторинг	Контроль состояния 2 блоков кода одновременно.

Элемент		**Описание**
OB	Цикл программы	Несколько
	Пуск	Несколько
	Прерывания по задержке времени	4 (1 на событие)
	Циклические прерывания	4 (1 на событие)
	Аппаратные прерывания	50 (1 на событие)
	Прерывания по ошибке времени	1
	Диагностические прерывания	1
	Извлечение или вставка модулей	1
	Ошибка стойки или станции	1
	Прерывание по времени	Несколько
	Состояние	1
	Обновление	1
	Профиль	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
	MC-PostServo	1
Таймеры	Тип	IEC
	Количество	Ограничивается только объемом памяти
	Сохранение	Структура в DB, 16 байт на таймер
Счетчики	Тип	IEC
	Количество	Ограничивается только объемом памяти
	Сохранение	Структура в DB, размер зависит от типа счёта • SInt, USInt: 3 байта • Int, UInt: 6 байт • DInt, UDInt: 12 байт

[1] Программы обеспечения безопасности используют два уровня вложенности. Таким образом, программа пользователя имеет в программах обеспечения безопасности четыре уровня вложенности.

Таблица A-70 Коммуникация

Технические данные	**Описание**
Интерфейсы	2
Тип	Ethernet
Устройство HMI	4
Программатор (PG)	1

Технические данные	Описание
Соединения	• 8 открытых коммуникационных соединений пользователя (активных или пассивных): TSEND_C, TRCV_C, TCON, TDISCON, TSEND и TRCV • 8 CPU/CPU соединений (клиент или сервер) для GET/PUT данных • 6 соединений для динамического распределения на GET/PUT или открытых коммуникационных соединений пользователя
Скорость передачи данных	10/100 Мбит/с
Электрическое разделение (между внешним сигналом и логикой)	Изолирующий трансформатор, 1500 В AC (типовое испытание)[1]
Тип кабеля	CAT5e экранированный
Интерфейсы	
Число интерфейсов PROFINET	1
Число интерфейсов PROFIBUS	0
Интерфейс	
Оборудование для интерфейса	
Число портов	2
Встроенный коммутатор	Да
RJ-45 (Ethernet)	Да; X1
Протоколы	
PROFINET IO контроллер	Да
PROFINET IO устройство	Да
Коммуникация SIMATIC	Да
Открытый обмен данными через IE	Да
Веб-сервер	Да
Резервирование среды	Да
PROFINET IO контроллер	
Службы	
Коммуникация PG/OP	Да
S7-маршрутизация	Да
Изохронный режим	Нет
Открытый обмен данными через IE	Да
IRT	Нет
MRP	Да; как MRP-клиент
PROFIenergy	Да. S7-1200 CPU поддерживает только объект PROFIenergy (с функциональностью I-устройства).
Пуск по приоритету	Да (до 16 устройств PROFINET)
Макс. число подключаемых IO-устройств	16
Макс. число подключаемых IO-устройств для RT	16
Из них в ряд, макс.	16
Макс. число одновременно активируемых/деактивируемых IO-устройств	8

Технические данные		Описание
Время обновления		Минимальное значение для времени обновления также зависит от настроенных для PROFINET IO коммуникационных компонентов, числа IO-устройств и количества сконфигурированных данных пользователя.
C RT		
Такт передачи в 1 мс		От 1 мс до 512 мс
PROFINET IO устройство		
Службы		
Коммуникация PG/OP		Да
S7-маршрутизация		Да
Изохронный режим		Нет
Открытый обмен данными через IE		Да
IRT, поддерживается		Нет
MRP, поддерживается		Да
PROFIenergy		Да
Устройство общего доступа		Да
Макс. количество IO-контроллеров с устройством общего доступа		2
Коммуникация SIMATIC		
S7-связь, как сервер		Да
S7-связь, как клиент		Да
Макс. данных пользователя за операцию		См. систему интерактивной помощи (S7-связь, объем данных пользователя)
Открытый обмен данными через IE		
TCP/IP:		Да
	Макс. длина данных	8 КБ
	Поддержка для нескольких пассивных соединений на порт	Да
ISO-on-TCP (RFC1006):		Да
	Макс. длина данных	8 КБ
UDP:		Да
	Макс. длина данных	1472 байта
DHCP		Нет
SNMP		Да
DCP		Да
LLDP		Да

[1] Изоляция порта Ethernet предназначена для снижение рисков из-за опасных напряжений при кратковременных сбоях сети. Она не соответствует требованиям безопасности для обычных развязок по напряжению в сетях переменного тока.

Таблица A-71 Электропитание

Технические данные		**CPU 1215C AC/DC/RLY**	**CPU 1215C DC/DC/RLY**	**CPU 1215C DC/DC/DC**
Диапазон напряжений		от 85 до 264 В AC	от 20,4 В DC до 28,8 В DC	
Частота сети		от 47 до 63 Гц	--	
Пусковой ток (макс. нагрузка)	Только CPU	100 мА при 120 В AC 50 мА при 240 В AC	500 мА при 24 В DC	
	CPU со всеми модулями расширения	300 мА при 120 В AC 150 мА при 240 В AC	1500 мА при 24 В DC	
Пусковой ток (макс.)		20 А при 264 В AC	12 А при 28,8 В DC	
I^2t		0,8 A^2 с	0,5 A^2 с	
Электрическое разделение (между внешним питанием и логикой)		1500 В AC	Без электрического разделения	
Ток утечки на землю, от линии переменного тока на функциональную землю		макс. 0,5 мА	-	
Допустимый перерыв в питании		20 мА при 120 В AC 80 мА при 240 В AC	10 мс при 24 В DC	
Встроенный предохранитель, недоступен для пользователя		3 А, 250 В, плавкий предохранитель		

Таблица A-72 Электропитание датчиков

Технические данные	**CPU 1215C AC/DC/RLY**	**CPU 1215C DC/DC/RLY**	**CPU 1215C DC/DC/DC**
Диапазон напряжений	от 20,4 до 28,8 В DC	L+ - 4 В DC (мин.)	
Номинальный выходной ток (макс.)	400 мА (с защитой от коротких замыканий)		
Пульсации напряжения (до 10 МГц), не более	< 1 В между пиками	Как в цепи питания	
Электрическая развязка (между логикой CPU и электропитанием датчиков)	Без электрического разделения		

A.7.3 Цифровые входы и выходы

Таблица A-73 Цифровые входы

Технические данные	**CPU 1215C AC/DC/RLY**	**CPU 1215C DC/DC/RLY**	**CPU 1215C DC/DC/DC**
Количество входов	14		
Тип	Потребитель/источник (IEC Тур 1, если потребитель)		
Номинальное напряжение	24 В DC при 4 мА, ном.		
Длительно допустимое напряжение	До 30 В DC		
Перенапряжение	35 В DC на 0,5 с		
Сигнал логический 1 (мин.)	15 В DC при 2,5 мА		
Сигнал логический 0 (макс.)	5 В DC при 1 мА		

Технические данные	**CPU 1215C AC/DC/RLY**	**CPU 1215C DC/DC/RLY**	**CPU 1215C DC/DC/DC**
Электрическое разделение (между полевой стороной и логикой)	707 В DC (типовое испытание)		
Изолированные группы	1		
Время фильтрации	мкс установка: 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0 мс установка: 0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0		
Скорость изменения входного сигнала для высокоскоростных счетчиков (макс.) (логический уровень 1 = от 15 до 26 В DC)	100/80 кГц (от Ia0 до Ia.5) 30/20 кГц (от Ia6 до Ib.5)		
Количество одновременно активных входов	• 7 (без смежных точек) при 60 °C для горизонтальной установки или 50 °C для вертикальной установки • 14 при 55 °C для горизонтальной установки или 45 °C для вертикальной установки		
Длина кабеля (в метрах)	500 м экранированный, 300 м неэкранированный, 50 м экранированный для входов высокоскоростных счетчиков		

Таблица A-74 Цифровые выходы

Технические данные	**CPU 1215C AC/DC/RLY и CPU 1215C DC/DC/RLY**	**CPU 1215C DC/DC/DC**
Выходы	10	
Тип	Замыкающие контакты реле	Транзисторные ключи MOSFET
Диапазон напряжений	от 5 до 30 В DC или от 5 до 250 В AC	от 20,4 до 28,8 В DC
Сигнал логический 1 при макс. токе	--	Мин. 20 В DC
Сигнал логический 0 при нагрузке 10 кОм	--	Макс. 0,1 В DC
Ток (макс.)	2,0 А	0,5 А
Ламповая нагрузка	30 Вт DC/200 Вт AC	5 Вт
Сопротивление в активном состоянии	Макс. 0,2 Ом для нового модуля	Макс. 0,6 Ом
Ток утечки на выход	--	До 10 мкА
Импульсный ток выхода, не более	7 А через замкнутый контакт	8 А макс. в течение 100 мс
Защита от перегрузки	Нет	
Электрическое разделение (между полевой стороной и логикой)	1500 В AC (катушка - контакт) Нет (катушка - логика)	707 В DC (типовое испытание)
Изолированные группы	2	1
Электрическая развязка (между группами)	1500 В AC [1]	--
Индуктивное фиксирующее напряжение	--	L+ - 48 В DC, 1 Вт рассеиваемая мощность
Задержка переключения (от Qa.0 до Qa.3)	До 10 мс	До 1,0 мкс из Выкл на Вкл До 3,0 мкс из Вкл на Выкл
Задержка переключения (от Qa.4 до Qb.1)	До 10 мс	До 5 мкс из Выкл на Вкл До 20 мкс из Вкл на Выкл

Технические данные	CPU 1215C AC/DC/RLY и CPU 1215C DC/DC/RLY	CPU 1215C DC/DC/DC
Макс. частота коммутации реле	1 Гц	--
Частота следования импульсов	Не рекомендуется [2]	100 кГц (от Qa.0 до Qa.3)[3], 2 Гц мин. 20 кГц (от Qa.4 до Qb.1)[3]
Механический срок службы (без нагрузки)	10.000.000 циклов переключения	--
Срок службы контактов при номинальной нагрузке	100.000 циклов переключения	--
Поведение при переходе из RUN в STOP	Последнее значение или замещающее значение (по умолчанию 0)	
Количество одновременно активных выходов	• 5 (без смежных точек) при 60 °C для горизонтальной установки или 50 °C для вертикальной установки • 10 при 55 °C для горизонтальной установки или 45 °C для вертикальной установки	
Длина кабеля (в метрах)	500 м экранированный, 150 м неэкранированный	

[1] Электрическое разделение между группами RLY развязывает сетевое напряжение и SELV/PELV, а также различные фазы до 250 В AC и линию заземления.

[2] Для вариантов CPU с релейными выходами для использования импульсных выходов должна быть установлена цифровая сигнальная плата (SB).

[3] В зависимости от используемого приемника импульсов и кабеля, дополнительный нагрузочный резистор (по крайней мере 10% от номинального тока) может улучшить качество импульсных сигналов и стойкость к воздействию помех.

A.7.4 Аналоговые входы и выходы

Таблица A-75 Аналоговые входы

Технические данные	Описание
Количество входов	2
Тип	Напряжение (асимметричный)
Диапазон изменения сигналов	от 0 до 10 В
Цифровое представление полной шкалы (слово данных)	от 0 до 27648
Выход за границы диапазона	от 10,001 до 11,759 В
Выход за границы диапазона (слово данных)	от 27649 до 32511
Переполнение	от 11,760 до 11,852 В
Переполнение (слово данных)	от 32512 до 32767
Разрешение	10 бит
Максимальное входное напряжение	35 В DC
Сглаживание	Отсутствует, слабое, среднее или сильное См. таблицу Реакция на скачок (мс) для аналоговых входов CPU (Страница 1493).
Подавление помех	10, 50 или 60 Гц
Полное сопротивление	≥100 кОм

Технические данные	Описание
Электрическое разделение (между полевой стороной и логикой)	Нет
Точность (25 °C / -20 до 60 °C)	3,0% / 3,5% по отношению к конечной точке шкалы
Длина кабеля (в метрах)	100 м, экранированная витая пара

A.7.4.1 Реакция на скачок встроенных аналоговых входов CPU

Таблица A-76 Реакция на скачок (мс), от 0 до 10 В измеренная при 95%

Выбор сглаживания (усреднение по выборке)	Частота подавления помех (время интегрирования)		
	60 Гц	**50 Гц**	**10 Гц**
Отсутствует (1 цикл): без усреднения	50 мс	50 мс	100 мс
Слабое (4 цикла): 4 значения	60 мс	70 мс	200 мс
Среднее (16 циклов) 16 значений	200 мс	240 мс	1150 мс
Сильное (32 цикла) 32 значения	400 мс	480 мс	2300 мс
Время выборки	**4,17 мс**	**5 мс**	**25 мс**

A.7.4.2 Время выборки для встроенных аналоговых портов CPU

Таблица A-77 Время выборки встроенных аналоговых входов CPU

Частота подавления помех (выбор времени интегрирования)	Время выборки
60 Гц (16,6 мс)	4,17 мс
50 Гц (20 мс)	5 мс
10 Гц (100 мс)	25 мс

A.7.4.3 Диапазоны измерений для аналоговых входов напряжения (CPU)

Таблица A-78 Представление аналогового входа для напряжения (CPU)

Система		Диапазон измерения напряжения	
Десятичная	**Шестнадцатеричная**	**от 0 до 10 В**	
32767	7FFF	11,852 В	Выход за верхний установленный предел
32512	7F00		
32511	7EFF	11,759 В	Выход за границы диапазона
27649	6C01		

Система		Диапазон измерения напряжения	
Десятичная	**Шестнадцате-ричная**	**от 0 до 10 В**	
27648	6C00	10 В	Расчетный диапазон
20736	5100	7,5 В	
34	22	12 мВ	
0	0	0 В	
Отрицательные значения		Отрицательные значения не поддерживаются	

A.7.4.4 Технические данные аналоговых выходов

Таблица A-79 Аналоговые выходы

Технические данные	Описание
Выходы	2
Тип	Ток
Диапазон изменения сигналов	0 ... 20 мА
Цифровое представление полной шкалы (слово данных)	от 0 до 27648
Выход за границы диапазона	от 20,01 до 23,52 мА
Выход за границы диапазона (слово данных)	от 27649 до 32511
Переполнение	См. примечание[1]
Переполнение (слово данных)	от 32512 до 32767
Разрешение	10 бит
Полное выходное сопротивление	≤ 500 Ом макс.
Электрическое разделение (между полевой стороной и логикой)	Нет
Точность (25 °C / -20 до 60 °C)	3,0% / 3,5% по отношению к конечной точке шкалы
Время установления	2 мс
Длина кабеля (в метрах)	100 м, экранированная витая пара

[1] При выходе за верхний установленный предел поведение аналоговых выходов соответствует характеристикам в конфигурации устройства. Выбрать для параметра "Реакция на CPU-STOP": "Подключить замещающее значение" или "Удерживать последнее значение".

Таблица A-80 Представление аналогового выхода для тока (CPU 1215C и CPU 1217C)

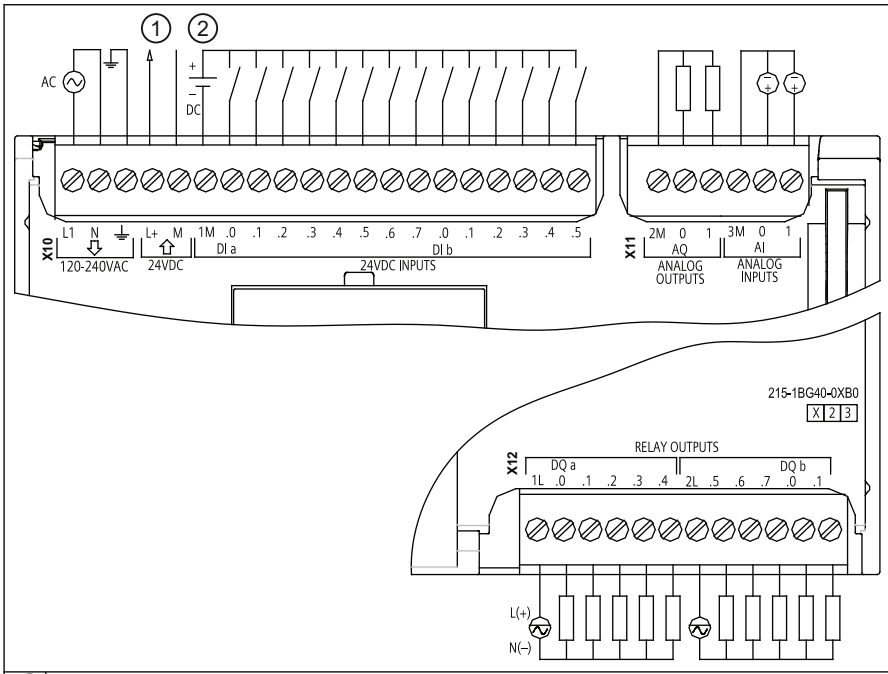
Система		Диапазон выходного тока	
Десятичная	**Шестнадцате-ричная**	**от 0 до 20 мА**	
32767	7FFF	См. Примечание 1	Выход за верхний установленный предел
32512	7F00	См. Примечание 1	
32511	7EFF	23,52 мА	Выход за границы диапазона
27649	6C01		

Система		Диапазон выходного тока	
Десятичная	**Шестнадцате-ричная**	**от 0 до 20 мА**	
27648	6C00	20 мА	Расчетный диапазон
20736	5100	15 мА	
34	22	0,0247 мА	
0	0	0 мА	
Отрицательные значения		Отрицательные значения не поддерживаются	

[1] При выходе за верхний установленный предел поведение аналоговых выходов соответствует установленным характеристикам в конфигурации устройства. Выбрать для параметра "Реакция на CPU-STOP": "Подключить замещающее значение" или "Удерживать последнее значение".

A.7.5 Схемы электрических соединений CPU 1215C

Таблица A-81 CPU 1215C AC/DC/RLY (6ES7215-1BG40-0XB0)



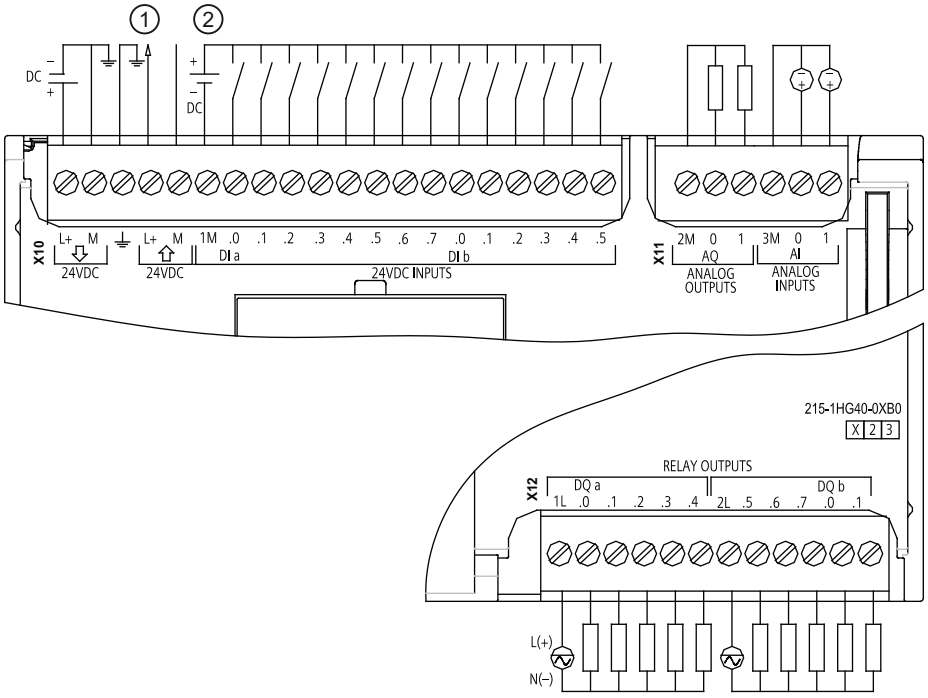
①	Выход питания датчиков 24 В DC Для дополнительной помехоустойчивости соединить "M" с корпусом, даже если питание датчика не используется.
②	Для входов-потребителей соединить "-" с "M" (см. рисунок). Для входов-источников соединить "+" с "M".
Примечание 1: Соединители X11 должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	

Примечание 2: Одна из двух клемм L1 или N (L2) может быть подключена к источнику напряжения до 240 В AC. Клемма N может рассматриваться как L2 и ее не нужно заземлять. Для клемм L1 и N (L2) не требуется защита от неправильной полярности.
Примечание 3: Дополнительную информацию о Ethernet порте CPU см. в Конфигурация устройства (Страница 151).

Таблица A-82 Разводка контактов для CPU 1215C AC/DC/RLY (6ES7215-1BG40-0XB0)

Контакт	X10	X11 (позолоченный)	X12
1	L1 / 120-240 В AC	2 M	1L
2	N / 120 - 240 В AC	AQ 0	DQ a.0
3	Функциональное заземление	AQ 1	DQ a.1
4	L+ / питание датчиков 24 В DC	3M	DQ a.2
5	M / питание датчиков 24 В DC	AI 0	DQ a.3
6	1M	AI 1	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Таблица A-83 CPU 1215C DC/DC/RLY (6ES7215-1HG40-0XB0)



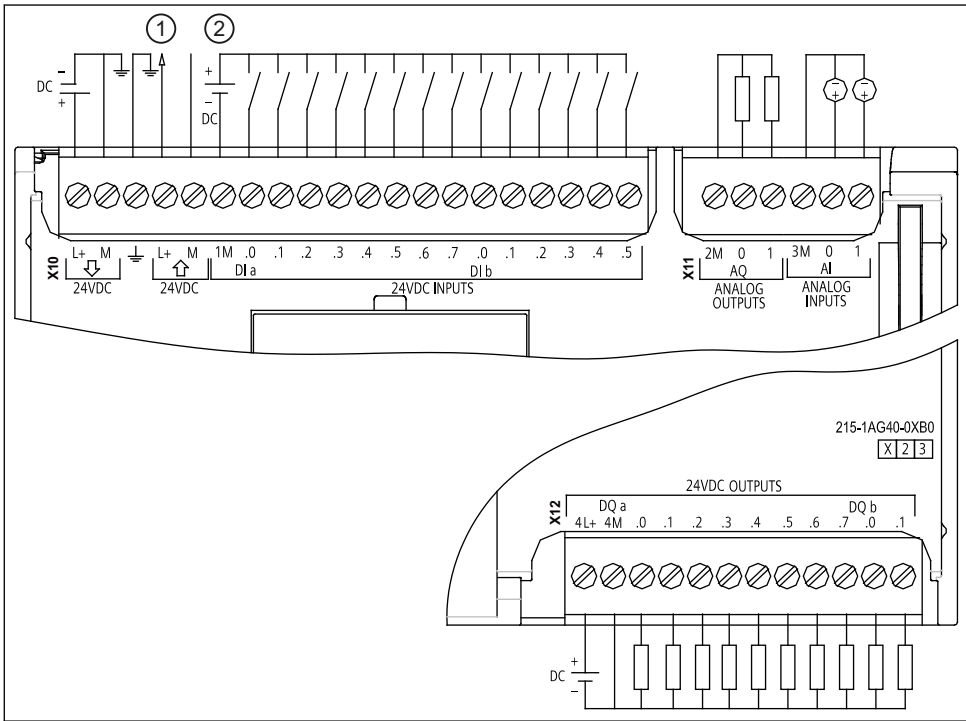
1	Выход питания датчиков 24 В DC Для дополнительной помехоустойчивости соединить "M" с корпусом, даже если питание датчика не используется.
2	Для входов-потребителей соединить "-" с "M" (см. рисунок). Для входов-источников соединить "+" с "M".
Примечание 1: Соединители X11 должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	
Примечание 2: Дополнительную информацию о Ethernet порте CPU см. в Конфигурация устройства (Страница 151).	

Таблица A-84 Разводка контактов для CPU 1215C DC/DC/RLY (6ES7215-1HG40-0XB0)

Контакт	X10	X11 (позолоченный)	X12
1	L+ / 24 В DC	2 M	1L
2	M / 24 В DC	AQ 0	DQ a.0
3	Функциональное заземление	AQ 1	DQ a.1
4	L+ / питание датчиков 24 В DC	3M	DQ a.2
5	M / питание датчиков 24 В DC	AI 0	DQ a.3
6	1M	AI 1	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0

Контакт	X10	X11 (позолоченный)	X12
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Таблица A-85 CPU 1215C DC/DC/DC (6ES7215-1AG40-0XB0)



①	Выход питания датчиков 24 В DC Для дополнительной помехоустойчивости соединить "M" с корпусом, даже если питание датчика не используется.
②	Для входов-потребителей соединить "-" с "M" (см. рисунок). Для входов-источников соединить "+" с "M".
Примечание 1: Соединители X11 должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	
Примечание 2: Дополнительную информацию о Ethernet порте CPU см. в Конфигурация устройства (Страница 151).	

Таблица A-86 Разводка контактов для CPU 1215C DC/DC/DC (6ES7215-1AG40-0XB0)

Контакт	X10	X11 (позолоченный)	X12
1	L1 / 24 В DC	2 M	4L+
2	M / 24 В DC	AQ 0	4M
3	Функциональное заземление	AQ 1	DQ a.0
4	L+ / питание датчиков 24 В DC	3M	DQ a.1
5	M / питание датчиков 24 В DC	AI 0	DQ a.2
6	1M	AI 1	DQ a.3
7	DI a.0	--	DQ a.4
8	DI a.1	--	DQ a.5
9	DI a.2	--	DQ a.6
10	DI a.3	--	DQ a.7
11	DI a.4	--	DQ b.0
12	DI a.5	--	DQ b.1
13	DI a.6	--	--
14	DI a.7	--	--
15	DI b.0	--	--
16	DI b.1	--	--
17	DI b.2	--	--
18	DI b.3	--	--
19	DI b.4	--	--
20	DI b.5	--	--

Примечание

Неиспользуемые аналоговые входы должны быть закорочены.

A.8 CPU 1217C

A.8.1 Общие технические данные и характеристики

Таблица A-87 Общее

Технические данные	CPU 1217C DC/DC/DC
Заказной номер	6ES7217-1AG40-0XB0
Габаритные размеры (Ш x В x Г) (мм)	150 x 100 x 75
Вес с упаковкой	530 г
Рассеиваемая мощность	12 Вт
Доступный электрический ток (SM и CM шина)	Макс. 1.600 мА (5 В DC)

Технические данные	CPU 1217C DC/DC/DC
Доступный электрический ток (24 В DC)	Макс. 400 мА (питание датчиков)
Потребление тока цифровым входом (24 В DC)	4 мА/вход

Таблица A-88 Характеристики CPU

Технические данные		Описание
Память пользователя (См. "Общие технические данные" (Страница 1433), "Сохранение во внутренней памяти CPU".)	Рабочая память	150 кБ
	Загружаемая память	4 МБ, внутренняя, расширяемая до объема SD-карты
	С сохранением	10 кБ
Встроенные цифровые I/O		14 входов/10 выходов
Встроенные аналоговые I/O		2 входа/2 выхода
Размер образа процесса		1024 байта для входов (I)/1024 байта для выходов (Q)
Битовая память (M)		8192 байта
Временная (локальная) память		• 16 кбайт для запуска и программного цикла, (включая соответствующие FB и FC) • 6 кбайт для каждого из других уровней приоритета прерываний (включая FB и FC)
Дополнительные сигнальные модули		Макс. 8 SM
Расширение с помощью SB, CB, BB		Макс. 1
Дополнительные коммуникационные модули		Макс. 3 CM
Высокоскоростные счётчики		До 6 сконфигурированных для использования любых встроенных или SB входов (см. Таблица аппаратного конфигурирования цифровых входов (DI) для CPU 1217C) (Страница 1505) • 1 МГц (от Ib.2 до Ib.5) • 100/[1]80 кГц (Ia.0 до Ia.5) • 30/[1]20 кГц (Ia.6 до Ib.1)
Импульсные выходы		До 4 сконфигурированных для использования любых встроенных или SB выходов (см. Таблица аппаратного конфигурирования цифровых выходов (DQ) для CPU 1217C) (Страница 1505). • 1 МГц (от Qa.0 до Qa.3) • 100 кГц (от Qa.4 до Qb.1)
Входы для захвата импульсов		14
Прерывания по задержке времени		4 с разрешением в 1 мс
Циклические прерывания		4 с разрешением в 1 мс
Прерывания по фронту		12 по нарастающему и 12 по спадающему (16 и 16 с дополнительной сигнальной платой)
Карта памяти		Карта памяти SIMATIC (опция)

Технические данные	Описание
Точность часов реального времени	+/- 60 сек в месяц
Буферизация часов реального времени	Тип. 20 дней/мин. 12 дней при 40 °C (необслуживаемый суперконденсатор)

[1] Более медленная скорость, если высокоскоростной счетчик сконфигурирован для квадратурного A/B режима работы.

Таблица A-89 Производительность

Тип инструкции		Скорость выполнения	
		Прямая адресация (I, Q и M)	Доступ к DB
Булево значение		0,08 мкс/инструкция	
Перемещение	Move_Bool	0,3 мкс/инструкция	1,17 мкс/инструкция
	Move_Word	0,137 мкс/инструкция	1,0 мкс/инструкция
	Move_Real	0,72 мкс/инструкция	1,0 мкс/инструкция
Математической операции сложения с плавающей запятой	Add Real	1,48 мкс/инструкция	1,78 мкс/инструкция

Примечание

Измеренное время зависит от множества факторов. Приведенные выше показатели производительности относятся к самым быстрым инструкциям в этой категории и к безошибочным программам.

A.8.2 Поддерживаемые CPU 1217C таймеры, счётчики и блоки кода

Таблица A-90 Поддерживаемые CPU 1217C блоки, таймеры и счётчики

Элемент		Описание
Блоки	Тип	OB, FB, FC, DB
	Размер	OB, FB, FC: 64 кбайт DB: до размера рабочей памяти
	Количество	В сумме до 1024 блоков (OB + FB + FC + DB)
	Диапазон адресов для FB, FC и DB	FB и FC: от 1 до 65535 (например, от FB 1 до FB 65535) DB: от 1 до 59999
	Глубина вложенности	16 для OB циклов или пусковых OB 6 для любых OB прерывающих событий[1]
	Мониторинг	Контроль состояния 2 блоков кода одновременно.

Элемент		Описание
OB	Цикл программы	Несколько
	Пуск	Несколько
	Прерывания по задержке времени	4 (1 на событие)
	Циклические прерывания	4 (1 на событие)
	Аппаратные прерывания	50 (1 на событие)
	Прерывания по ошибке времени	1
	Диагностические прерывания	1
	Извлечение или вставка модулей	1
	Ошибка стойки или станции	1
	Прерывание по времени	Несколько
	Состояние	1
	Обновление	1
	Профиль	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
	MC-PostServo	1
Таймеры	Тип	IEC
	Количество	Ограничивается только объемом памяти
	Сохранение	Структура в DB, 16 байт на таймер
Счетчики	Тип	IEC
	Количество	Ограничивается только объемом памяти
	Сохранение	Структура в DB, размер зависит от типа счёта • SInt, USInt: 3 байта • Int, UInt: 6 байт • DInt, UDInt: 12 байт

[1] Программы обеспечения безопасности используют два уровня вложенности. Таким образом, программа пользователя имеет в программах обеспечения безопасности четыре уровня вложенности.

Таблица A-91 Коммуникация

Технические данные	Описание
Интерфейсы	2
Тип	Ethernet
Устройство HMI	4
Программатор (PG)	1

Технические данные	Описание
Соединения	• 8 открытых коммуникационных соединений пользователя (активных или пассивных): TSEND_C, TRCV_C, TCON, TDISCON, TSEND и TRC • 8 CPU/CPU соединений (клиент или сервер) для GET/PUT данных • 6 соединений для динамического распределения на GET/PUT или открытых коммуникационных соединений пользователя
Скорость передачи данных	10/100 Мбит/с
Электрическое разделение (между внешним сигналом и логикой)	Изолирующий трансформатор, 1500 В AC (типовое испытание)[1]
Тип кабеля	CAT5e экранированный
Интерфейсы	
Число интерфейсов PROFINET	1
Число интерфейсов PROFIBUS	0
Интерфейс	
Оборудование для интерфейса	
Число портов	2
Встроенный коммутатор	Да
RJ-45 (Ethernet)	Да; X1
Протоколы	
PROFINET IO контроллер	Да
PROFINET IO устройство	Да
Коммуникация SIMATIC	Да
Открытый обмен данными через IE	Да
Веб-сервер	Да
Резервирование среды	Да
PROFINET IO контроллер	
Службы	
Коммуникация PG/OP	Да
S7-маршрутизация	Да
Изохронный режим	Нет
Открытый обмен данными через IE	Да
IRT	Нет
MRP	Да; как MRP-клиент
PROFIenergy	Да. S7-1200 CPU поддерживает только объект PROFIenergy (с функциональностью I-устройства).
Пуск по приоритету	Да (до 16 устройств PROFINET)
Макс. число подключаемых IO-устройств	16
Макс. число подключаемых IO-устройств для RT	16
Из них в ряд, макс.	16
Макс. число одновременно активируемых/деактивируемых IO-устройств	8

Технические данные		Описание
Время обновления		Минимальное значение для времени обновления также зависит от настроенных для PROFINET IO коммуникационных компонентов, числа IO-устройств и количества сконфигурированных данных пользователя.
C RT		
Такт передачи в 1 мс		От 1 мс до 512 мс
PROFINET IO устройство		
Службы		
Коммуникация PG/OP		Да
S7-маршрутизация		Да
Изохронный режим		Нет
Открытый обмен данными через IE		Да
IRT, поддерживается		Нет
MRP, поддерживается		Да
PROFIenergy		Да
Устройство общего доступа		Да
Макс. количество IO-контроллеров с устройством общего доступа		2
Коммуникация SIMATIC		
S7-связь, как сервер		Да
S7-связь, как клиент		Да
Макс. данных пользователя за операцию		См. систему интерактивной помощи (S7-связь, объем данных пользователя)
Открытый обмен данными через IE		
TCP/IP:		Да
	Макс. длина данных	8 КБ
	Поддержка для нескольких пассивных соединений на порт	Да
ISO-on-TCP (RFC1006):		Да
	Макс. длина данных	8 КБ
UDP:		Да
	Макс. длина данных	1472 байта
DHCP		Нет
SNMP		Да
DCP		Да
LLDP		Да

[1] Изоляция порта Ethernet предназначена для снижение рисков из-за опасных напряжений при кратковременных сбоях сети. Она не соответствует требованиям безопасности для обычных развязок по напряжению в сетях переменного тока.

Таблица A-92 Электропитание

Технические данные	CPU 1217C DC/DC/DC
Диапазон напряжений	от 20,4 В DC до 28,8 В DC
Частота сети	--

Технические данные		**CPU 1217C DC/DC/DC**
Входной ток (макс. нагрузка)	Только CPU	600 мА при 24 В DC
	CPU со всеми модулями расширения	1600 мА при 24 В DC
Пусковой ток (макс.)		12 А при 28,8 В DC
$I^2 t$		0,5 A^2 с
Электрическое разделение (между внешним питанием и логикой)		Без электрического разделения
Допустимый перерыв в питании		10 мс при 24 В DC
Встроенный предохранитель, недоступен для пользователя		3 А, 250 В, плавкий предохранитель

Таблица A-93 Электропитание датчиков

Технические данные	**CPU 1217C DC/DC/DC**
Диапазон напряжений	L+ - 4 В DC (мин.)
Номинальный выходной ток (макс.)	400 мА (с защитой от коротких замыканий)
Пульсации напряжения (до 10 МГц), не более	Как в цепи питания
Электрическая развязка (между логикой CPU и электропитанием датчиков)	Без электрического разделения

A.8.3 Цифровые входы и выходы

Таблица A-94 Цифровые входы

Технические данные	**CPU 1217C DC/DC/DC**
Количество входов	14: Всего: 10: Потребитель/источник (IEC Typ 1, если потребитель) 4: Дифференциальный (RS422/RS485)
Тип: Потребитель/источник (IEC Typ 1 потребитель)	от Ia0 до Ia.7, от Ib.0 до Ib.1
Номинальное напряжение	24 В DC при 4 мА, ном.
Длительно допустимое напряжение	До 30 В DC
Перенапряжение	35 В DC на 0,5 с
Сигнал логический 1 (мин.)	15 В DC при 2,5 мА
Сигнал логический 0 (макс.)	5 В DC при 1 мА
Электрическое разделение (между полевой стороной и логикой)	707 В DC (типовое испытание)
Изолированные группы	1
Время фильтрации	мкс установка: 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0 мс установка: 0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0

Технические данные	CPU 1217C DC/DC/DC
Скорость изменения входного сигнала для высокоскоростных счетчиков (макс.) (логический уровень 1 = от 15 до 26 В DC)	100/80 кГц (от Ia0 до Ia.5) 30/20 кГц (от Ia6 до Ib.1)
Тип: Дифференциальный вход (RS422/RS485)	от Ib.2 до Ib.5 (0,2+ 0,2- до 0,5+ 0,5-)
Диапазон синфазного напряжения	от -7 до +12 В в течение 1 сек., 3 В длительно (характеристика RS422/RS485)
Встроенная концевая нагрузка и смещение	390 Ом на 2M от Ib"-", 390 Ом на +5 В от Ib"-", (выключено при разомкнутой цепи T/B) 220 Ом между Ib"+" и Ib"-"
Входное сопротивление приемника	100 Ом, включая терминальные резисторы и смещение
Дифференциальный порог чувствительность приемника	Не менее ±0,2 В, типовой гистерезис 60 мВ (характеристика RS422/RS485)
Электрическое разделение (между полевой стороной и логикой)	707 В DC (типовое испытание)
Изолированные группы	1
Время фильтрации	мкс установка: 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0 мс установка: 0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0
Скорость изменения входного сигнала высокоскоростных счетчиков (макс.)	Частота следования 1-фазных входных сигналов: 1 МГц (от Ib.2 до Ib.5) Частота следования 2-фазных входных сигналов 1 МГц (от Ib.2 до Ib.5)
Перекос между каналами дифференциального входа	До 40 нс
Общие технические данные (все цифровые входы)	
Количество одновременно активных входов	5 потребитель/источник (без смежных точек) и 4 дифференциальных входа при 60 °C для горизонтальной установки или 50 °C для вертикальной установки 14 при 55 °C для горизонтальной установки или 45 °C для вертикальной установки
Длина кабеля (в метрах)	500 м экранированный, 300 м неэкранированный 50 м экранированный для входов высокоскоростных счетчиков (потребитель/источник) 50 м экранированная витая пара для всех дифференциальных входов

Таблица A-95 Таблица аппаратного конфигурирования цифровых входов (DI) для CPU 1217C

Вход	Тип и скорость
DIa.0	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: макс. 100 кГц
DIa.1	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: макс. 100 кГц
DIa.2	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: макс. 100 кГц
DIa.3	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: макс. 100 кГц

Вход	Тип и скорость
DIa.4	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: макс. 100 кГц
DIa.5	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: макс. 100 кГц
DIa.6	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: макс. 30 кГц
DIa.7	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: макс. 30 кГц
DIb.0	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: макс. 30 кГц
DIb.1	Тип: 24 В, вход потребитель-источник тип 1 Скорость изменения входного сигнала: макс. 30 кГц
DIb.2+ .2-	Тип: RS422/RS485 дифференциальных вход Скорость изменения входного сигнала: макс. 1 МГц
DIb.3+ .3-	Тип: RS422/RS485 дифференциальных вход Скорость изменения входного сигнала: макс. 1 МГц
DIb.4+ .4-	Тип: RS422/RS485 дифференциальных вход Скорость изменения входного сигнала: макс. 1 МГц
DIb.5+ .5-	Тип: RS422/RS485 дифференциальных вход Скорость изменения входного сигнала: макс. 1 МГц

Таблица A-96 Цифровые выходы

Технические данные	**CPU 1217C DC/DC/DC**
Выходы	10 всего 6: Транзисторные ключи MOSFET 4: Дифференциальный (RS422/RS485)
Тип: Транзисторные ключи MOSFET (выход-источник)	Qa.4 до Qb.1
Диапазон напряжений	от 20,4 до 28,8 В DC
Сигнал логический 1 при макс. токе	Мин. 20 В DC
Сигнал логический 0 при нагрузке 10 кОм	Макс. 0,1 В DC
Ток (макс.)	0,5 А
Ламповая нагрузка	5 Вт
Сопротивление в активном состоянии	Макс. 0,6 Ом
Ток утечки на выход	До 10 мкА
Импульсный ток выхода, не более	8 А макс. в течение 100 мс
Защита от перегрузки	Нет
Электрическое разделение (между полевой стороной и логикой)	707 В DC (типовое испытание)

Технические данные	CPU 1217C DC/DC/DC
Изолированные группы	1
Индуктивное фиксирующее напряжение	L+ - 48 В DC, 1 Вт рассеиваемая мощность
Задержка переключения (от Qa.4 до Qb.1)	До 1,0 мкс из Выкл на Вкл До 3,0 мкс из Вкл на Выкл
Макс. частота коммутации реле	--
Частота следования импульсов	до 100 кГц (Qa.4 до Qb.1)[1], мин. 2 Гц
Тип: Дифференциальный выход (RS422/RS485)	от Qa.0 до Qa.3 (от .0+ .0- до .3+ .3-)
Диапазон синфазного напряжения	от -7 до +12 В в течение 1 сек., 3 В длительно (характеристика RS422/RS485)
Дифференциальное выходное напряжение передатчика	мин. 2 В при RL = 100 Ом, мин. 1,5 В при RL = 54 Ом (характеристика RS422/RS485)
Встроенная концевая нагрузка	100 Ом между Qa'+' и Qa'-'
Выходное сопротивление драйвера	100 Ом, включая терминальный резистор
Электрическое разделение (между полевой стороной и логикой)	707 В DC (типовое испытание)
Изолированные группы	1
Задержка переключения (DQa.0 до DQa.3)	До 100 нс
Перекос между каналами дифференциального выхода	До 40 нс
Частота следования импульсов	1 МГц (от Qa.0 до Qa.3), мин. 2 Гц
Общие технические данные (все цифровые выходы)	
Поведение при переходе из RUN в STOP	Последнее значение или замещающее значение (по умолчанию 0)
Количество одновременно активных выходов	3 выхода MOSFET, электронные (источник) (без смежных точек) и 4 дифференциальных выхода при 60 °C для горизонтальной установки или 50 °C для вертикальной установки 10 при 55 °C для горизонтальной установки или 45 °C для вертикальной установки
Длина кабеля (в метрах)	500 м экранированный, 150 м неэкранированный

[1] В зависимости от используемого приемника импульсов и кабеля, дополнительный нагрузочный резистор (по крайней мере 10% от номинального тока) может улучшить качество импульсных сигналов и стойкость к воздействию помех.

Таблица A-97 Таблица аппаратного конфигурирования цифровых выходов (DQ) для CPU 1217C

Выход	Тип и скорость
DQa.0+ .0-	Тип: Дифференциальный выход RS422/RS485 Частота следования импульсов: макс. 1 МГц, мин. 2 Гц
DQa.1+ .1-	Тип: Дифференциальный выход RS422/RS485 Частота следования импульсов: макс. 1 МГц, мин. 2 Гц
DQa.2+ .2-	Тип: Дифференциальный выход RS422/RS485 Частота следования импульсов: макс. 1 МГц, мин. 2 Гц

Выход	Тип и скорость
DQa.3+ .3-	Тип: Дифференциальный выход RS422/RS485 Частота следования импульсов: макс. 1 МГц, мин. 2 Гц
DQa.4	Тип: 24 В выход-источник Частота следования импульсов: макс. 100 кГц, мин. 2 Гц
DQa.5	Тип: 24 В выход-источник Частота следования импульсов: макс. 100 кГц, мин. 2 Гц
DQa.6	Тип: 24 В выход-источник Частота следования импульсов: макс. 100 кГц, мин. 2 Гц
DQa.7	Тип: 24 В выход-источник Частота следования импульсов: макс. 100 кГц, мин. 2 Гц
DQb.0	Тип: 24 В выход-источник Частота следования импульсов: макс. 100 кГц, мин. 2 Гц
DQb.1	Тип: 24 В выход-источник Частота следования импульсов: макс. 100 кГц, мин. 2 Гц

A.8.4 Аналоговые входы и выходы

A.8.4.1 Технические данные аналоговых входов

Таблица A-98 Аналоговые входы

Технические данные	Описание
Количество входов	2
Тип	Напряжение (асимметричный)
Диапазон изменения сигналов	от 0 до 10 В
Цифровое представление полной шкалы (слово данных)	от 0 до 27648
Выход за границы диапазона	от 10,001 до 11,759 В
Выход за границы диапазона (слово данных)	от 27649 до 32511
Переполнение	от 11,760 до 11,852 В
Переполнение (слово данных)	от 32512 до 32767
Разрешение	10 бит
Максимальное входное напряжение	35 В DC
Сглаживание	Отсутствует, слабое, среднее или сильное См. таблицу Реакция на скачок (мс) для аналоговых входов CPU (Страница 1510).
Подавление помех	10, 50 или 60 Гц
Полное сопротивление	≥100 кОм
Электрическое разделение (между полевой стороной и логикой)	Нет
Точность (25 °C / -20 до 60 °C)	3,0% / 3,5% по отношению к конечной точке шкалы
Длина кабеля (в метрах)	100 м, экранированная витая пара

A.8.4.2 Реакция на скачок встроенных аналоговых входов CPU

Таблица A-99 Реакция на скачок (мс), от 0 до 10 В измеренная при 95%

Выбор сглаживания (усреднение по выборке)	**Частота подавления помех (время интегрирования)**		
	60 Гц	**50 Гц**	**10 Гц**
Отсутствует (1 цикл): без усреднения	50 мс	50 мс	100 мс
Слабое (4 цикла): 4 значения	60 мс	70 мс	200 мс
Среднее (16 циклов) 16 значений	200 мс	240 мс	1150 мс
Сильное (32 цикла) 32 значения	400 мс	480 мс	2300 мс
Время выборки	**4,17 мс**	**5 мс**	**25 мс**

A.8.4.3 Время выборки для встроенных аналоговых портов CPU

Таблица A-100 Время выборки встроенных аналоговых входов CPU

Частота подавления помех (выбор времени интегрирования)	**Время выборки**
60 Гц (16,6 мс)	4,17 мс
50 Гц (20 мс)	5 мс
10 Гц (100 мс)	25 мс

A.8.4.4 Диапазоны измерений для аналоговых входов напряжения (CPU)

Таблица A-101 Представление аналогового входа для напряжения (CPU)

Система		**Диапазон измерения напряжения**	
Десятичная	**Шестнадцатеричная**	**от 0 до 10 В**	
32767	7FFF	11,852 В	Выход за верхний установленный предел
32512	7F00		
32511	7EFF	11,759 В	Выход за границы диапазона
27649	6C01		
27648	6C00	10 В	Расчетный диапазон
20736	5100	7,5 В	
34	22	12 мВ	
0	0	0 В	
Отрицательные значения		Отрицательные значения не поддерживаются	

A.8.4.5 Технические данные аналоговых выходов

Таблица A-102 Аналоговые выходы

Технические данные	Описание
Выходы	2
Тип	Ток
Диапазон изменения сигналов	0 ... 20 мА
Цифровое представление полной шкалы (слово данных)	от 0 до 27648
Выход за границы диапазона	от 20,01 до 23,52 мА
Выход за границы диапазона (слово данных)	от 27649 до 32511
Переполнение	См. примечание[1]
Переполнение (слово данных)	от 32512 до 32767
Разрешение	10 бит
Полное выходное сопротивление	≤ 500 Ом макс.
Электрическое разделение (между полевой стороной и логикой)	Нет
Точность (25 °C / -20 до 60 °C)	3,0% / 3,5% по отношению к конечной точке шкалы
Время установления	2 мс
Длина кабеля (в метрах)	100 м, экранированная витая пара

[1] При выходе за верхний установленный предел поведение аналоговых выходов соответствует характеристикам в конфигурации устройства. Выбрать для параметра "Реакция на CPU-STOP": "Подключить замещающее значение" или "Удерживать последнее значение".

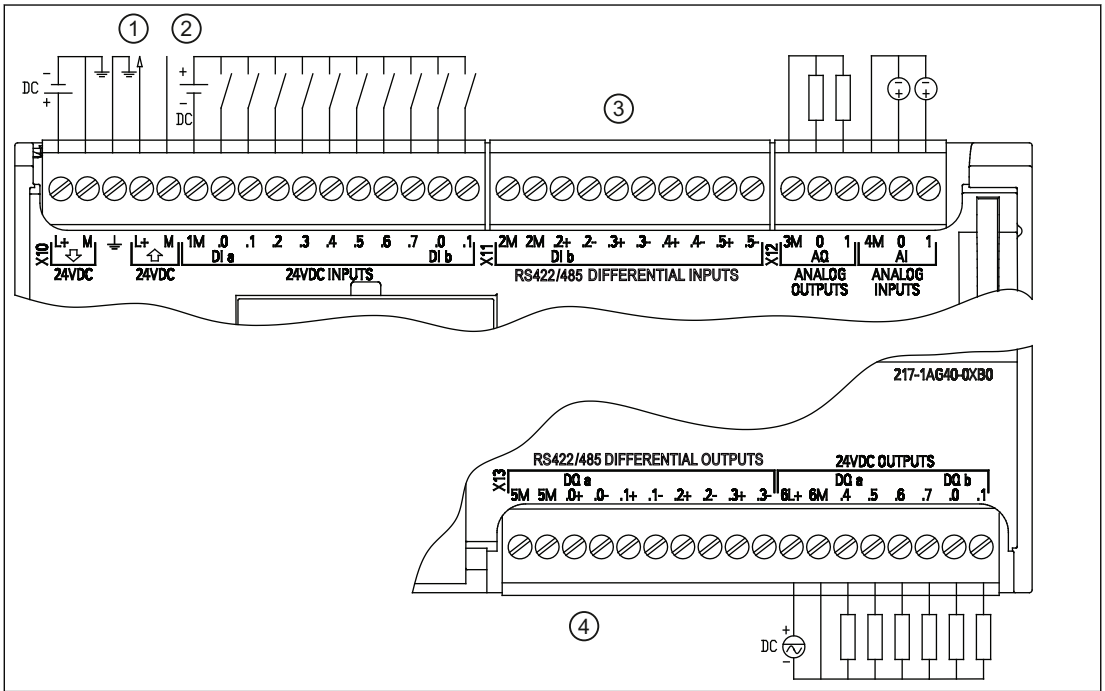
Таблица A-103 Представление аналогового выхода для тока (CPU 1215C и CPU 1217C)

Система		Диапазон выходного тока	
Десятичная	**Шестнадцате-ричная**	**от 0 до 20 мА**	
32767	7FFF	См. Примечание 1	Выход за верхний установленный предел
32512	7F00	См. Примечание 1	
32511	7EFF	23,52 мА	Выход за границы диапазона
27649	6C01		
27648	6C00	20 мА	Расчетный диапазон
20736	5100	15 мА	
34	22	0,0247 мА	
0	0	0 мА	
Отрицательные значения		Отрицательные значения не поддерживаются	

[1] При выходе за верхний установленный предел поведение аналоговых выходов соответствует установленным характеристикам в конфигурации устройства. Выбрать для параметра "Реакция на CPU-STOP": "Подключить замещающее значение" или "Удерживать последнее значение".

A.8.5 Схемы электрических соединений CPU 1217C

Таблица A-104 CPU 1217C DC/DC/DC (6ES7217-1AG40-0XB0)



①	Выход питания датчиков 24 В DC Для дополнительной помехоустойчивости соединить "M" с корпусом, даже если питание датчика не используется.
②	Для входов-потребителей соединить "-" с "M" (см. рисунок). Для входов-источников соединить "+" с "M".
③	См. Подробные сведения и примеры использования дифференциального входа (DI) для CPU 1217C (Страница 1514).
④	См. Подробные сведения и примеры использования дифференциального выхода (DQ) для CPU 1217C (Страница 1516).
Примечание 1: Соединители X12 должны быть позолоченными. См. Приложение C, запасные части (Страница 1642) для заказного номера.	
Примечание 2: Дополнительную информацию о Ethernet порте CPU см. в Конфигурация устройства (Страница 151).	

Таблица A-105 Разводка контактов для CPU 1217C DC/DC/DC (6ES7217-1AG40-0XB0)

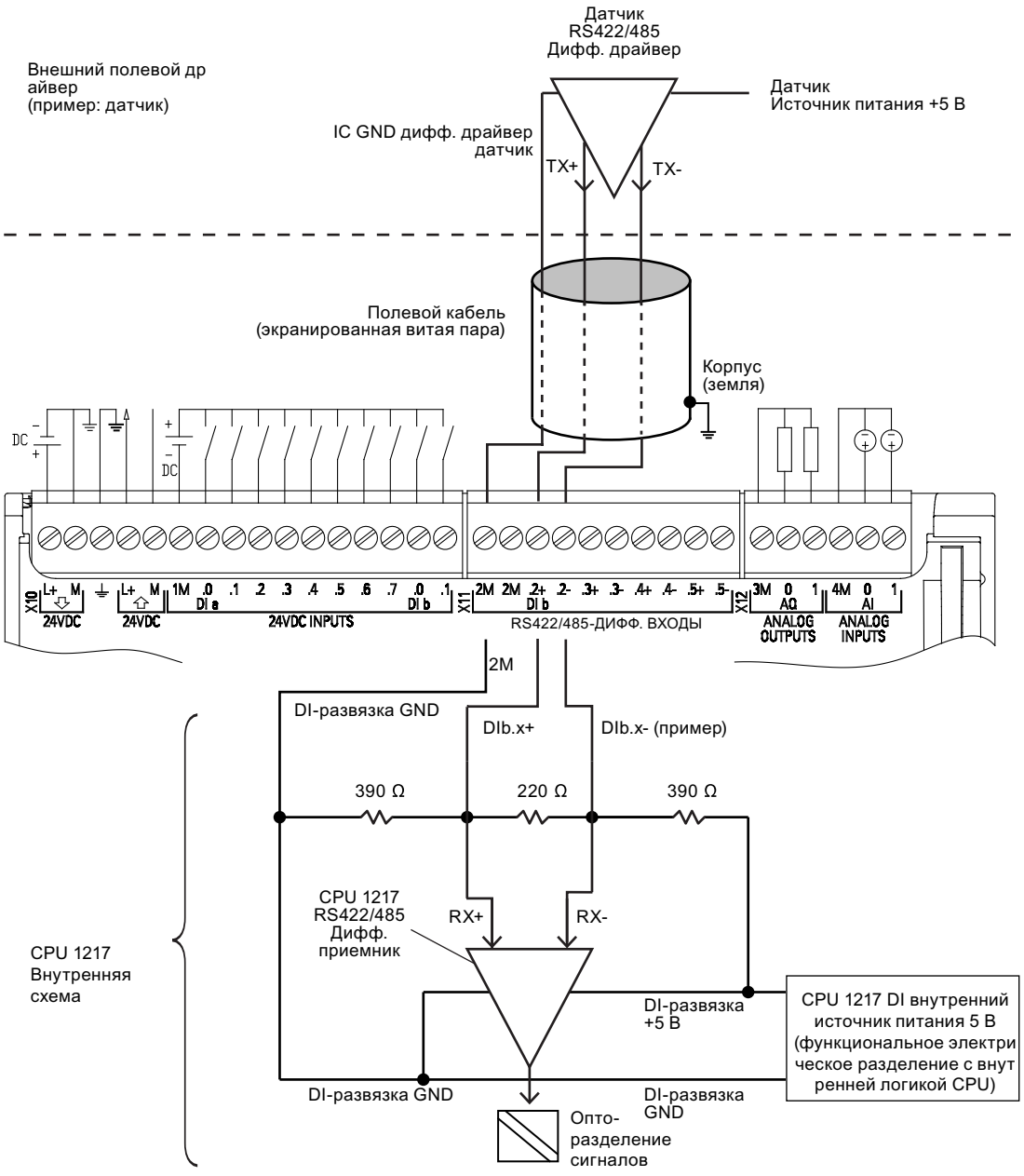
Контакт	X10	X11	X12 (позолоченный)	X13
1	L+ / 24 В DC	2M	3M	5M
2	M / 24 В DC	2M	AQ 0	5M
3	Функциональное заземление	DI b.2+	AQ 1	DQ a.0+

Контакт	X10	X11	X12 (позолоченный)	X13
4	L+ / питание датчиков 24 В DC	DI b.2-	4M	DQ a.0-
5	M / питание датчиков 24 В DC	DI b.3+	AI 0	DQ a.1+
6	1M	DI b.3-	AI 1	DQ a.1-
7	DI a.0	DI a.4+	--	DQ a.2+
8	DI a.1	DI b.4-	--	DQ a.2-
9	DI a.2	DI b.5+	--	DQ a.3+
10	DI a.3	DI b.5-	--	DQ a.3-
11	DI a.4	--	--	6L+
12	DI a.5	--	--	6M
13	DI a.6	--	--	DQ a.4
14	DI a.7	--	--	DQ a.5
15	DI b.0	--	--	DQ a.6
16	DI b.1	--	--	DQ a.7
17	--	--	--	DQ b.0
18	--	--	--	DQ b.1

Примечание

Неиспользуемые аналоговые входы должны быть закорочены.

A.8.6 Подробные сведения и примеры использования дифференциального входа (DI) для CPU 1217C

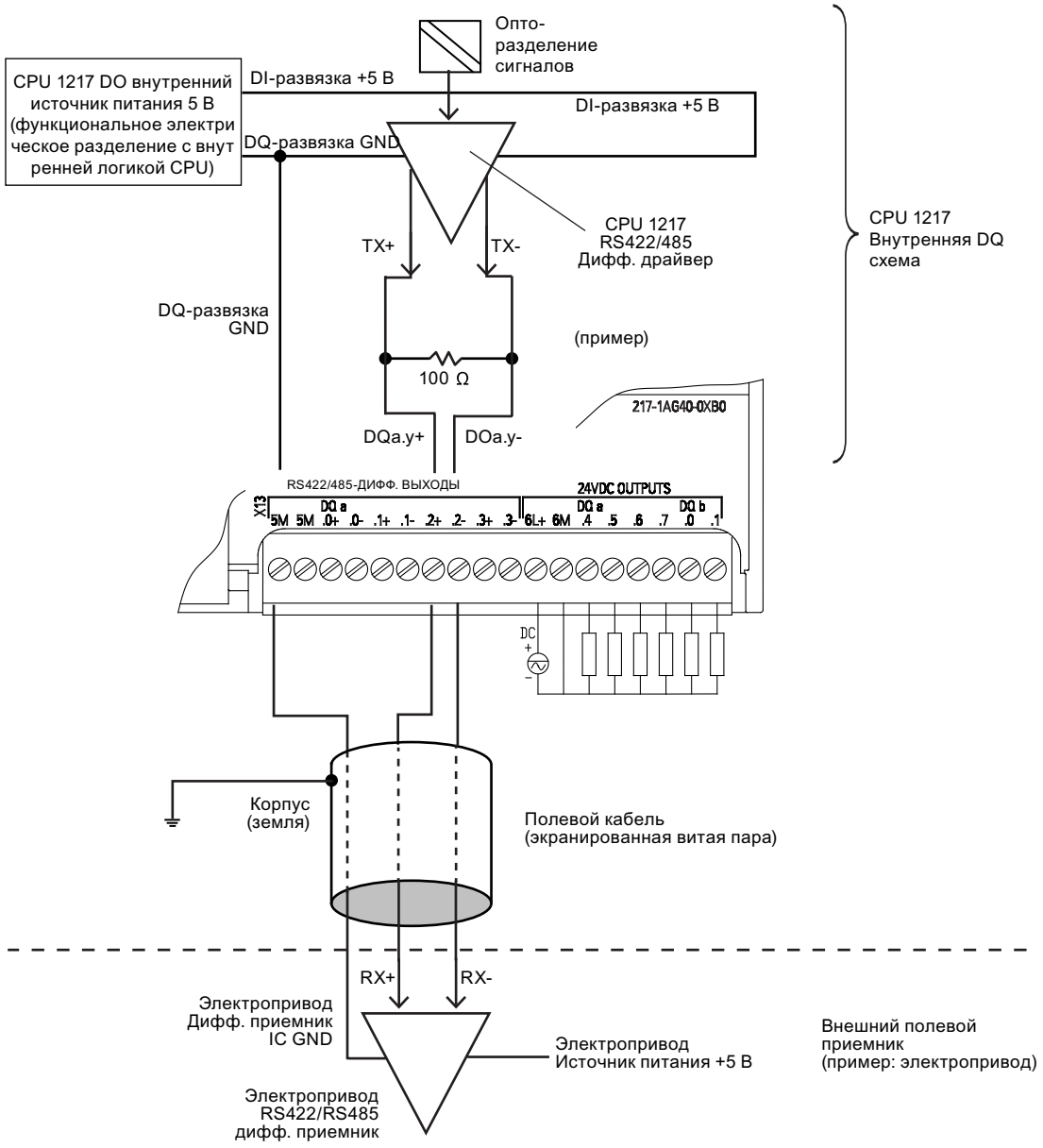


Примечания

- Каждый дифференциальный цифровой вход смещен в состояние "ВЫКЛ.", когда винтовые зажимы на клеммной колодке разомкнуты.
- Встроенные оконечная нагрузка и смещение DI имеют эквивалентный импеданс 100 Ом.

- Диапазон синфазного напряжения ограничивается встроенным DI терминальным резистором и резисторами в цепи смещения. Подробную информацию можно найти в Технических данных.

A.8.7 Подробные сведения и примеры использования дифференциального выхода (DQ) для CPU 1217C



Ука-
за-
ние

- Диапазон синфазного напряжения ограничивается встроенным DI терминальным резистором. Подробную информацию можно найти в Технических данных.

A.9 Цифровые сигнальные модули (SM)

A.9.1 Технические данные для модуля цифрового ввода SM 1221

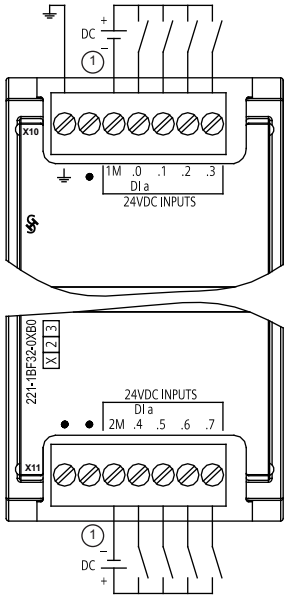
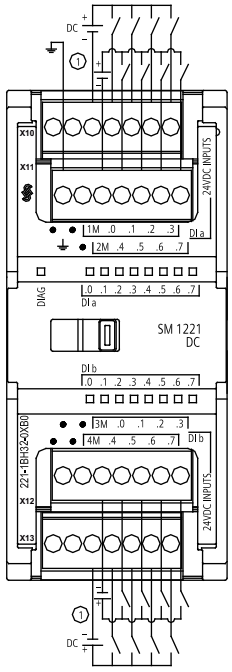
Таблица A-106 Общие технические данные

Модель	**SM 1221 DI 8 x 24 VDC**	**SM 1221 DI 16 x 24 VDC**
Заказной номер	6ES7221-1BF32-0XB0	6ES7221-1BH32-0XB0
Габаритные размеры (Ш x В x Г) (мм)	45 x 100 x 75	
Вес	170 г	210 г
Рассеиваемая мощность	1,5 Вт	2,5 Вт
Потребляемый ток (SM шина)	105 мА	130 мА
Потребляемый ток (24 В DC)	4 мА / вход	

Таблица A-107 Цифровые входы

Модель	**SM 1221 DI 8 x 24 VDC**	**SM 1221 DI 16 x 24 VDC**
Количество входов	8	16
Тип	Потребитель/источник (IEC Typ 1, если потребитель)	
Номинальное напряжение	24 В DC при 4 мА, ном.	
Длительно допустимое напряжение	До 30 В DC	
Перенапряжение	35 В DC на 0,5 с	
Сигнал логический 1 (мин.)	15 В DC при 2,5 мА	
Сигнал логический 0 (макс.)	5 В DC при 1 мА	
Электрическое разделение (между полевой стороной и логикой)	707 В DC (типовое испытание)	
Изолированные группы	2	4
Время фильтрации	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 и 12,8 мс (настраивается для каждой группы из 4 входных каналов)	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 и 12,8 мс (настраивается для каждой группы из 4 входных каналов)
Количество одновременно активных входов	8	16
Длина кабеля (в метрах)	500 м экранированный, 300 м неэкранированный	

Таблица A-108 Схемы электрических соединений для SM цифрового ввода

SM 1221 DI 8 x 24 VDC (6ES7221-1BF32-0XB0)	SM 1221 DI 16 x 24 VDC (6ES7221-1BH32-0XB0)
	

① Для входов-потребителей соединить "-" с "M" (см. рисунок). Для входов-источников соединить "+" с "M".

Таблица A-109 Разводка контактов для SM 1221 DI 8 x 24 VDC (6ES7221-1BF32-0XB0)

Контакт	X10	X11
1	Функциональное заземление	Не подключено
2	Не подключено	Не подключено
3	1M	2M
4	DI a.0	DI a.4
5	DI a.1	DI a.5
6	DI a.2	DI a.6
7	DI a.3	DI a.7

Таблица A-110 Разводка контактов для SM 1221 DI 16 x 24 VDC (6ES7221-1BH32-0XB0)

Контакт	X10	X11	X12	X13
1	Не подключено	Функциональное заземление	Не подключено	Не подключено
2	Не подключено	Не подключено	Не подключено	Не подключено
3	1M	2M	3 M	4 M
4	DI a.0	DI a.4	DI b.0	DI b.4
5	DI a.1	DI a.5	DI b.1	DI b.5
6	DI a.2	DI a.6	DI b.2	DI b.6
7	DI a.3	DI a.7	DI b.3	DI b.7

A.9.2 Технические данные для модуля цифрового вывода SM 1222 с 8 выходами

Таблица A-111 Общие технические данные

Модель	**SM 1222 DQ 8 x Relay**	**SM 1222 DQ 8 x Relay (переключение)**	**SM 1222 DQ 8 x 24 VDC**
Заказной номер	6ES7222-1HF32-0XB0	6ES7222-1XF32-0XB0	6ES7222-1BF32-0XB0
Габаритные размеры (Ш x В x Г) (мм)	45 x 100 x 75	70 x 100 x 75	45 x 100 x 75
Вес	190 г	310 г	180 г
Рассеиваемая мощность	4,5 Вт	5 Вт	1,5 Вт
Потребляемый ток (SM шина)	120 мА	140 мА	120 мА
Потребляемый ток (24 В DC)	11 мА / катушка реле	16,7 мА / катушка реле	50 мА

Таблица A-112 Цифровые выходы

Модель	**SM 1222 DQ 8 x Relay**	**SM 1222 DQ 8 x Relay (переключение)**	**SM 1222 DQ 8 x 24 VDC**
Выходы	8	8	8
Тип	Замыкающие контакты реле	Переключающий контакт реле	Транзисторные ключи MOSFET
Диапазон напряжений	от 5 до 30 В DC или от 5 до 250 В AC	от 5 до 30 В DC или от 5 до 250 В AC	от 20,4 до 28,8 В DC
Сигнал логический 1 при макс. токе	--	--	Мин. 20 В DC
Сигнал логический 0 при нагрузке 10 кОм	--	--	Макс. 0,1 В DC
Ток (макс.)	2,0 А	2,0 А	0,5 А
Ламповая нагрузка	30 Вт DC/200 Вт AC	30 Вт DC/200 Вт AC	5 Вт
Сопротивление активного выходного канала	До 0,2 Ом для нового контакта	До 0,2 Ом для нового контакта	Макс. 0,6 Ом

Модель	**SM 1222 DQ 8 x Relay**	**SM 1222 DQ 8 x Relay (переключение)**	**SM 1222 DQ 8 x 24 VDC**
Ток утечки на выход	--	--	До 10 мкА
Импульсный ток выхода, не более	7 А через замкнутый контакт	7 А через замкнутый контакт	8 А макс. в течение 100 мс
Защита от перегрузки	Нет	Нет	
Электрическое разделение (между полевой стороной и логикой)	1500 В AC (катушка - контакт) Без (катушка - логика)	1500 В AC (катушка - контакт)	707 В DC (типовое испытание)
Изолированные группы	2	8	1
Ток на общий провод (макс.)	10 А	2 А	4 А
Индуктивное фиксирующее напряжение	--	--	L+ - 48 В, 1 Вт рассеиваемая мощность
Задержка переключения	До 10 мс	До 10 мс	До 50 мкс из Выкл на Вкл До 200 мкс из Вкл на Выкл
Макс. частота коммутации реле	1 Гц	1 Гц	--
Механический срок службы (без нагрузки)	10.000.000 циклов переключения	10.000.000 циклов переключения	--
Срок службы контактов при номинальной нагрузке (замыкающий контакт)	100.000 циклов переключения	100.000 циклов переключения	--
Поведение при переходе из RUN в STOP	Последнее значение или замещающее значение (по умолчанию 0)	Последнее значение или замещающее значение (по умолчанию 0)	Последнее значение или замещающее значение (по умолчанию 0)
Количество одновременно активных выходов	8	• 4 (без смежных точек) при 60 °C для горизонтальной установки или 50 °C для вертикальной установки • 8 при 55 °C для горизонтальной установки или 45 °C для вертикальной установки	8
Длина кабеля (в метрах)	500 м экранированный, 150 м неэкранированный	500 м экранированный, 150 м неэкранированный	500 м экранированный, 150 м неэкранированный

A.9.3 Технические данные для модуля цифрового вывода SM 1222 с 16 выходами

Таблица A-113 Общие технические данные

Модель	**SM 1222 DQ 16 x Relay**	**SM 1222 DQ 16 x 24 VDC**	**SM 1222 DQ 16 x 24 VDC потребитель**
Заказной номер	6ES7222-1HH32-0XB0	6ES7222-1BH32-0XB0	6ES7222-1BH32-1XB0
Габаритные размеры (Ш x В x Г) (мм)	45 x 100 x 75	45 x 100 x 75	45 x 100 x 75
Вес	260 г	220 г	220 г
Рассеиваемая мощность	8,5 Вт	2,5 Вт	2,5 Вт
Потребляемый ток (SM шина)	135 мА	140 мА	140 мА
Потребляемый ток (24 В DC)	11 мА / катушка реле	100 мА	40 мА

Таблица A-114 Цифровые выходы

Модель	**SM 1222 DQ 16 x Relay**	**SM 1222 DQ 16 x 24 VDC**	**SM 1222 DQ 16 x 24 VDC потребитель**
Выходы	16	16	16
Тип	Замыкающие контакты реле	Транзисторные ключи MOSFET	Транзисторные ключи MOSFET
Диапазон напряжений	от 5 до 30 В DC или от 5 до 250 В AC	от 20,4 до 28,8 В DC	от 20,4 до 28,8 В DC
Сигнал логический 1 при макс. токе	-	Мин. 20 В DC	0,5 В DC
Сигнал логический 0 при нагрузке 10 кОм	-	Макс. 0,1 В DC	24 В (тип.) минус 0,75 В DC
Ток (макс.)	2,0 А	0,5 А	0,5 А
Ламповая нагрузка	30 Вт DC/200 Вт AC	5 Вт	5 Вт
Сопротивление активного выходного канала	До 0,2 Ом для нового контакта	Макс. 0,6 Ом	Макс. 0,5 Ом
Ток утечки на выход	--	До 10 мкА	До 75 мкА
Импульсный ток выхода, не более	7 А через замкнутый контакт	8 А макс. в течение 100 мс	8 А макс. в течение 100 мс
Защита от перегрузки	Нет		Да, защищенный ограничителем тока диапазон от 1 А до 3,5 А
Электрическое разделение (между полевой стороной и логикой)	1500 В AC (катушка - контакт) Без (катушка - логика)	707 В DC (типовое испытание)	707 В DC (типовое испытание)
Изолированные группы	4	1	1
Ток на общий провод (макс.)	10 А	8 А	Ограничение тока
Индуктивное фиксирующее напряжение	-	L+ - 48 В, 1 Вт рассеиваемая мощность	45 В

Модель	**SM 1222 DQ 16 x Relay**	**SM 1222 DQ 16 x 24 VDC**	**SM 1222 DQ 16 x 24 VDC потребитель**
Задержка переключения	До 10 мс	До 50 мкс из Выкл на Вкл До 200 мкс из Вкл на Выкл	До 20 мкс из Выкл на Вкл До 350 мкс из Вкл на Выкл
Макс. частота коммутации реле	1 Гц	-	-
Механический срок службы (без нагрузки)	10.000.000 циклов переключения	-	-
Срок службы контактов при номинальной нагрузке (замыкающий контакт)	100.000 циклов переключения	-	-
Поведение при переходе из RUN в STOP	Последнее значение или замещающее значение (по умолчанию 0)	Последнее значение или замещающее значение (по умолчанию 0)	Последнее значение или замещающее значение (по умолчанию 0)
Количество одновременно активных выходов	• 8 (без смежных точек) при 60 °C для горизонтальной установки или 50 °C для вертикальной установки • 16 при 55 °C для горизонтальной установки или 45 °C для вертикальной установки	16	16
Длина кабеля (в метрах)	500 м экранированный, 150 м неэкранированный		

См. также

Новые функции (Страница 35)

Таблица A-115 Схемы электрических соединений для SM цифрового вывода с 8 выходами

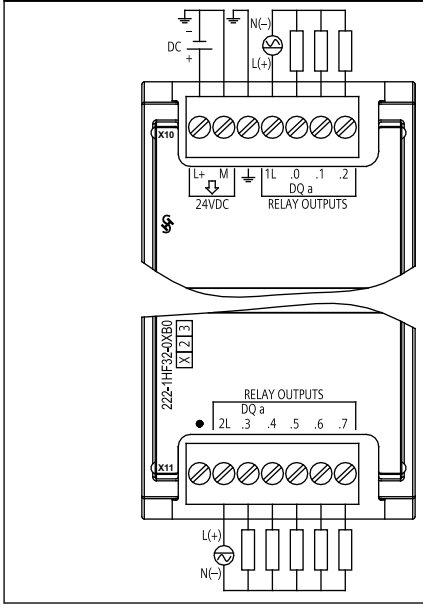
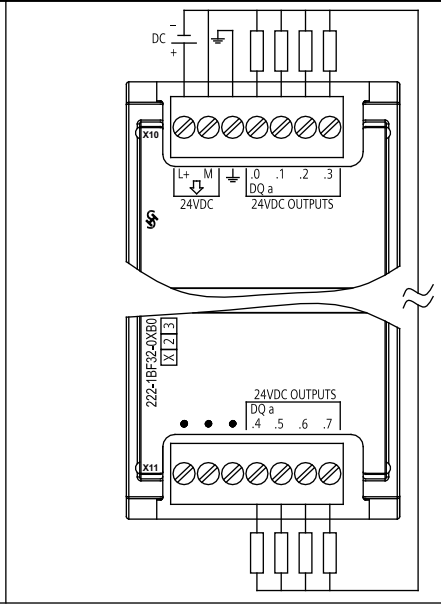
SM 1222 DQ 8 x Relay (6ES7222-1HF32-0XB0)	**SM 1222 DQ 8 x 24 VDC (6ES7222-1BF32-0XB0)**
	

Таблица A-116 Разводка контактов для SM 1222 DQ 8 x Relay (6ES7222-1HF32-0XB0)

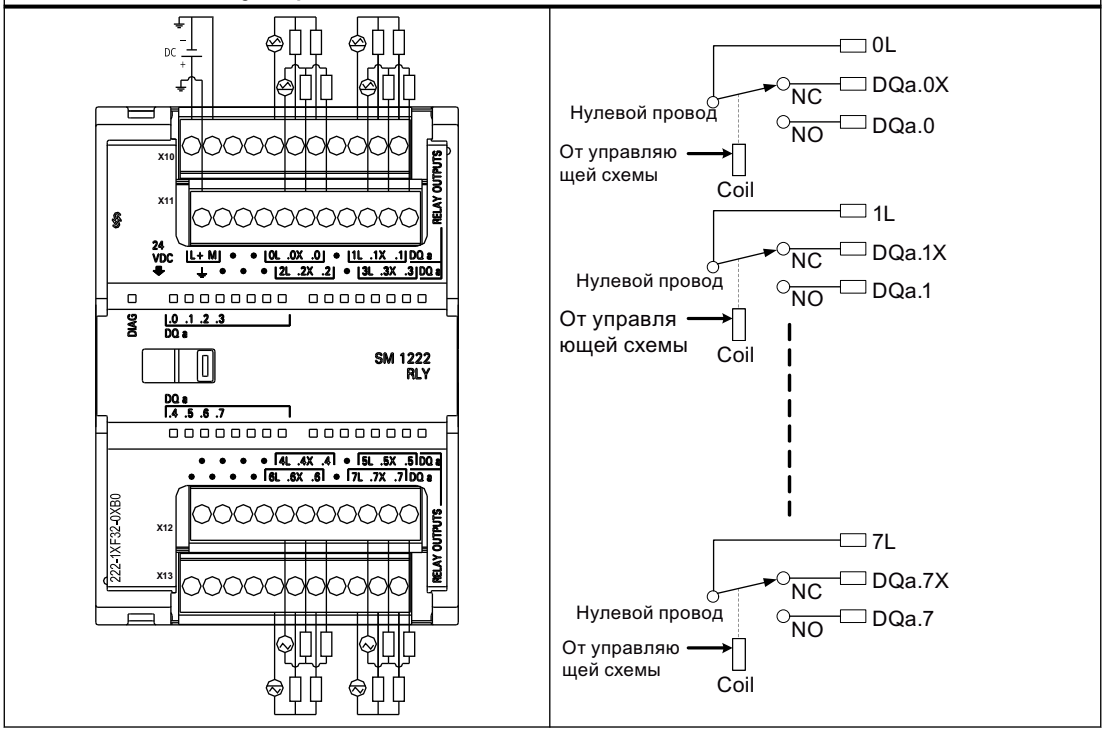
Контакт	X10	X11
1	L+ / 24 В DC	Не подключено
2	M / 24 В DC	2L
3	Функциональное заземление	DQ a.3
4	1L	DQ a.4
5	DQ a.0	DQ a.5
6	DQ a.1	DQ a.6
7	DQ a.2	DQ a.7

Таблица A-117 Разводка контактов для SM 1222 DQ 8 x 24 VDC (6ES7222-1BF32-0XB0)

Контакт	X10	X11
1	L+ / 24 В DC	Не подключено
2	M / 24 В DC	Не подключено
3	Функциональное заземление	Не подключено
4	DQ a.0	DQ a.4
5	DQ a.1	DQ a.5
6	DQ a.2	DQ a.6
7	DQ a.2	DQ a.7

Таблица A-118 Схема электрических соединений для SM цифрового вывода с 8 релейными выходами с переключающим контактом

SM 1222 DQ 8 x Relay, переключение (6ES7222-1XF32-0XB0)



Переключающий релейный выход управляет двумя цепями с помощью соединения с общим проводом: одного нормально замкнутого и одного нормально разомкнутого контакта. Например, с помощью выхода "0", в состоянии OFF, общий провод (0L) соединяется с нормально замкнутым контактом (.0X) и отключается от нормально разомкнутого контакта (.0). В состоянии выхода ON общий провод (0L) отключается нормально замкнутого контакта (.0X) и соединяется с нормально разомкнутым контактом (.0).

Таблица A-119 Разводка контактов для SM 1222 DQ 8 x Relay, переключение (6ES7222-1XF32-0XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В DC	Функциональное заземление	Не подключено	Не подключено
2	M / 24 В DC	Не подключено	Не подключено	Не подключено
3	Не подключено	Не подключено	Не подключено	Не подключено
4	Не подключено	Не подключено	Не подключено	Не подключено
5	0L	2L	4L	6L
6	DQ a.0X	DQ a.2X	DQ a.4X	DQ a.6X
7	DQ a.0	DQ a.2	DQ a.4	DQ a.6
8	Не подключено	Не подключено	Не подключено	Не подключено
9	1L	3L	5L	7L
10	DQ a.1X	DQ a.3X	DQ a.5X	DQ a.7X
11	DQ a.1	DQ a.3	DQ a.5	DQ a.7

Таблица A-120 Схемы электрических соединений для SM цифрового вывода с 16 выходами

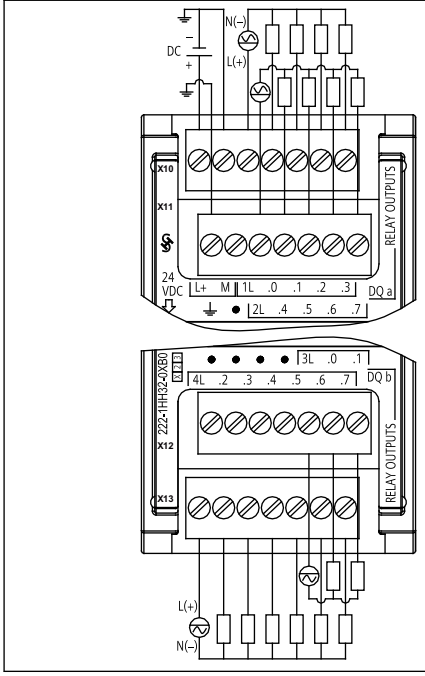
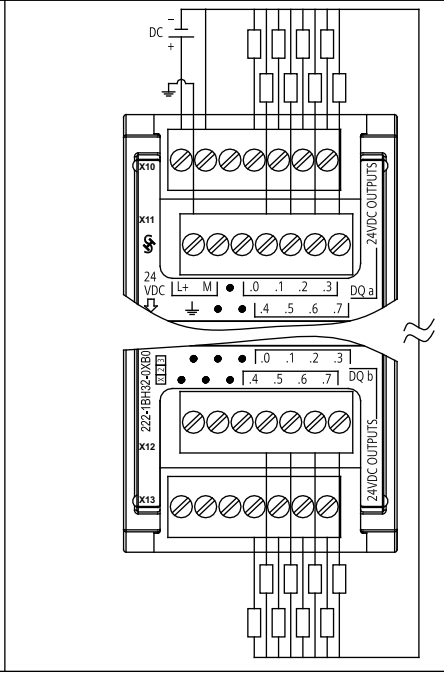
SM 1222 DQ 16 x Relay (6ES7222-1HH32-0XB0)	**SM 1222 DQ 16 x 24 VDC (6ES7222-1BH32-0XB0)**
	

Таблица A-121 Разводка контактов для SM 1222 DQ 16 x Relay (6ES7222-1HH32-0XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В DC	Функциональное заземление	Не подключено	4L
2	M / 24 В DC	Не подключено	Не подключено	DQ b.2
3	1L	2L	Не подключено	DQ b.3
4	DQ a.0	DQ a.4	Не подключено	DQ b.4
5	DQ a.1	DQ a.5	3L	DQ b.5
6	DQ a.2	DQ a.6	DQ b.0	DQ b.6
7	DQ a.3	DQ a.7	DQ b.1	DQ b.7

Таблица A-122 Разводка контактов для SM 1222 DQ 16 x 24 VDC (6ES7222-1BH32-0XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В DC	Функциональное заземление	Не подключено	Не подключено
2	M / 24 В DC	Не подключено	Не подключено	Не подключено
3	Не подключено	Не подключено	Не подключено	Не подключено

Контакт	X10	X11	X12	X13
4	DQ a.0	DQ a.4	DQ b.0	DQ b.4
5	DQ a.1	DQ a.5	DQ b.1	DQ b.5
6	DQ a.2	DQ a.6	DQ b.2	DQ b.6
7	DQ a.3	DQ a.7	DQ b.3	DQ b.7

Таблица A-123 Схема электрических соединений для SM цифрового вывода с 16 выходами, 24 В DC потребитель

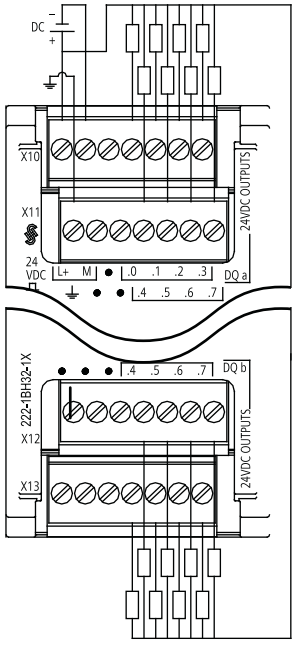
SM 1222 DQ 16 x 24 VDC потребитель (6ES7222-1BH32-1XB0)	
	

Таблица A-124 Разводка контактов для SM 1222 DQ 16 x 24 VDC потребитель (6ES7222-1BH32-1XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В DC	Функциональное заземление	Не подключено	Не подключено
2	M / 24 В DC	Не подключено	Не подключено	Не подключено
3	Не подключено	Не подключено	Не подключено	Не подключено
4	DQ a.0	DQ a.4	DQ b.0	DQ b.4
5	DQ a.1	DQ a.5	DQ b.1	DQ b.5
6	DQ a.2	DQ a.6	DQ b.2	DQ b.6
7	DQ a.3	DQ a.7	DQ b.3	DQ b.7

A.9.4 Технические данные для модуля цифрового ввода/вывода постоянного тока SM 1223

Таблица A-125 Общие технические данные

Модель	**SM 1223 DI 8 x 24 VDC, DQ 8 x Relay**	**SM 1223 DI 16 x 24 VDC, DQ 16 x Relay**	**SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC**	**SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC**	**SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC потребитель**
Заказной номер	6ES7223-1PH32-0XB0	6ES7223-1PL32-0XB0	6ES7223-1BH32-0XB0	6ES7223-1BL32-0XB0	6ES7223-1BL32-1XB0
Габаритные размеры (Ш x В x Г) (мм)	45 x 100 x 75	70 x 100 x 75	45 x 100 x 75	70 x 100 x 75	70 x 100 x 75
Вес	230 г	350 г	210 г	310 г	310 г
Рассеиваемая мощность	5,5 Вт	10 Вт	2,5 Вт	4,5 Вт	4,5 Вт
Потребляемый ток (SM шина)	145 мА	180 мА	145 мА	185 мА	185 мА
Потребляемый ток (24 В DC)	4 мА / вход 11 мА / катушка реле		150 мА	200 мА	40 мА

Таблица A-126 Цифровые входы

Модель	**SM 1223 DI 8 x 24 VDC, DQ 8 x Relay**	**SM 1223 DI 16 x 24 VDC, DQ 16 x Relay**	**SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC**	**SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC**	**SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC потребитель**
Количество входов	8	16	8	16	16
Тип	Потребитель/ источник (IEC Тур 1 потребитель)	Потребитель/ источник (IEC Тур 1 потребитель)	Потребитель/ источник (IEC Тур 1 потребитель)	Потребитель/ источник (IEC Тур 1 потребитель)	Потребитель/ источник (IEC Тур 1 потребитель)
Номинальное напряжение	24 В DC при 4 мА, ном.	24 В DC при 4 мА, ном.	24 В DC при 4 мА, ном.	24 В DC при 4 мА, ном.	24 В DC при 4 мА, ном.
Длительно допустимое напряжение	До 30 В DC	До 30 В DC	До 30 В DC	До 30 В DC	До 30 В DC
Перенапряжение	35 В DC на 0,5 с	35 В DC на 0,5 с	35 В DC на 0,5 с	35 В DC на 0,5 с	35 В DC на 0,5 с
Сигнал логический 1 (мин.)	15 В DC при 2,5 мА	15 В DC при 2,5 мА	15 В DC при 2,5 мА	15 В DC при 2,5 мА	15 В DC при 2,5 мА
Сигнал логический 0 (макс.)	5 В DC при 1 мА	5 В DC при 1 мА	5 В DC при 1 мА	5 В DC при 1 мА	5 В DC при 1 мА
Электрическое разделение (между полевой стороной и логикой)	707 В DC (типовое испытание)	707 В DC (типовое испытание)	707 В DC (типовое испытание)	707 В DC (типовое испытание)	707 В DC (типовое испытание)
Изолированные группы	2	2	2	2	2

Модель	**SM 1223 DI 8 x 24 VDC, DQ 8 x Relay**	**SM 1223 DI 16 x 24 VDC, DQ 16 x Relay**	**SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC**	**SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC**	**SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC потребитель**
Время фильтрации	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 и 12,8 мс (настраивается для каждой группы из 4 входных каналов)	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 и 12,8 мс (настраивается для каждой группы из 4 входных каналов)	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 и 12,8 мс (настраивается для каждой группы из 4 входных каналов)	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 и 12,8 мс (настраивается для каждой группы из 4 входных каналов)	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 и 12,8 мс (настраивается для каждой группы из 4 входных каналов)
Количество одновременно активных входов	8	• 8 (без смежных точек) при 60 °C для горизонтальной установки или 50 °C для вертикальной установки • 16 при 55 °C для горизонтальной установки или 45 °C для вертикальной установки	8	16	16
Длина кабеля (в метрах)	500 м экранированный, 300 м неэкранированный	500 м экранированный, 300 м неэкранированный	500 м экранированный, 300 м неэкранированный	500 м экранированный, 300 м неэкранированный	500 м экранированный, 300 м неэкранированный

Таблица A-127 Цифровые выходы

Модель	**SM 1223 DI 8 x 24 VDC, DQ 8 x Relay**	**SM 1223 DI 16 x 24 VDC, DQ 16 x Relay**	**SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC**	**SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC**	**SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC потребитель**
Выходы	8	16	8	16	16
Тип	Замыкающие контакты реле	Замыкающие контакты реле	Транзисторные ключи MOSFET	Транзисторные ключи MOSFET	Транзисторные ключи MOSFET
Диапазон напряжений	от 5 до 30 В DC или от 5 до 250 В AC	от 5 до 30 В DC или от 5 до 250 В AC	от 20,4 до 28,8 В DC	от 20,4 до 28,8 В DC	от 20,4 до 28,8 В DC
Сигнал логический 1 при макс. токе	--	--	Мин. 20 В DC	Мин. 20 В DC	0,5 В DC
Сигнал логический 0 при нагрузке 10 кОм	--	--	Макс. 0,1 В DC	Макс. 0,1 В DC	24 В (тип.) минус 0,75 В DC
Ток (макс.)	2,0 A	2,0 A	0,5 A	0,5 A	0,5 A

Модель	**SM 1223 DI 8 x 24 VDC, DQ 8 x Relay**	**SM 1223 DI 16 x 24 VDC, DQ 16 x Relay**	**SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC**	**SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC**	**SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC потребитель**
Ламповая нагрузка	30 Вт DC/200 Вт AC	30 Вт DC/200 Вт AC	5 Вт	5 Вт	5 Вт
Сопротивление активного выходного канала	До 0,2 Ом для нового контакта	До 0,2 Ом для нового контакта	Макс. 0,6 Ом	Макс. 0,6 Ом	Макс. 0,5 Ом
Ток утечки на выход	--	--	До 10 мкА	До 10 мкА	До 75 мкА
Импульсный ток выхода, не более	7 А через замкнутый контакт	7 А через замкнутый контакт	8 А макс. в течение 100 мс	8 А макс. в течение 100 мс	Ограничение тока
Защита от перегрузки	Нет	Нет	Нет	Нет	Да, защищенный ограничителем тока диапазон от 1 А до 3,5 А
Электрическое разделение (между полевой стороной и логикой)	1500 В AC (катушка - контакт) Без (катушка - логика)	1500 В AC (катушка - контакт) Без (катушка - логика)	707 В DC (типовое испытание)	707 В DC (типовое испытание)	707 В DC (типовое испытание)
Изолированные группы	2	4	1	1	1
Ток на общий провод	10 А	8 А	4 А	8 А	8 А
Индуктивное фиксирующее напряжение	--	--	L+ - 48 В, 1 Вт рассеиваемая мощность	L+ - 48 В, 1 Вт рассеиваемая мощность	45 В
Задержка переключения	До 10 мс	До 10 мс	До 50 мкс из Выкл на Вкл До 200 мкс из Вкл на Выкл	До 50 мкс из Выкл на Вкл До 200 мкс из Вкл на Выкл	До 20 мкс из Выкл на Вкл До 350 мкс из Вкл на Выкл
Макс. частота коммутации реле	1 Гц	1 Гц	--	--	--
Механический срок службы (без нагрузки)	10.000.000 циклов переключения	10.000.000 циклов переключения	--	--	--
Срок службы контактов при номинальной нагрузке (замыкающий контакт)	100.000 циклов переключения	100.000 циклов переключения	--	--	--
Поведение при переходе из RUN в STOP	Последнее значение или замещающее значение (по умолчанию 0)	Последнее значение или замещающее значение (по умолчанию 0)	Последнее значение или замещающее значение (по умолчанию 0)	Последнее значение или замещающее значение (по умолчанию 0)	Последнее значение или замещающее значение (по умолчанию 0)

Модель	**SM 1223 DI 8 x 24 VDC, DQ 8 x Relay**	**SM 1223 DI 16 x 24 VDC, DQ 16 x Relay**	**SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC**	**SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC**	**SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC потребитель**
Количество одновременно активных выходов	8	• 8 (без смежных точек) при 60 °C для горизонтальной установки или 50 °C для вертикальной установки • 16 при 55 °C для горизонтальной установки или 45 °C для вертикальной установки	8	16	16
Длина кабеля (в метрах)	500 м экранированный, 150 м неэкранированный	500 м экранированный, 150 м неэкранированный	500 м экранированный, 150 м неэкранированный	500 м экранированный, 150 м неэкранированный	500 м экранированный, 150 м неэкранированный

См. также

Новые функции (Страница 35)

Таблица A-128 Схемы электрических соединений для модулей цифрового ввода/вывода (VDC/Relay)

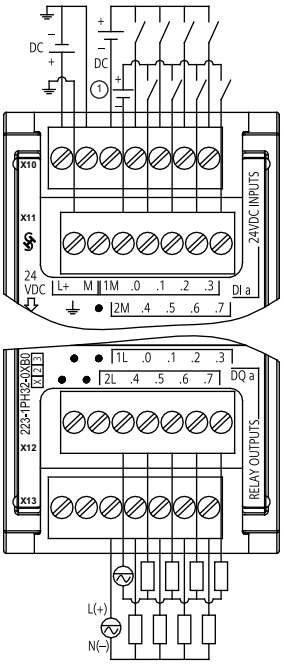
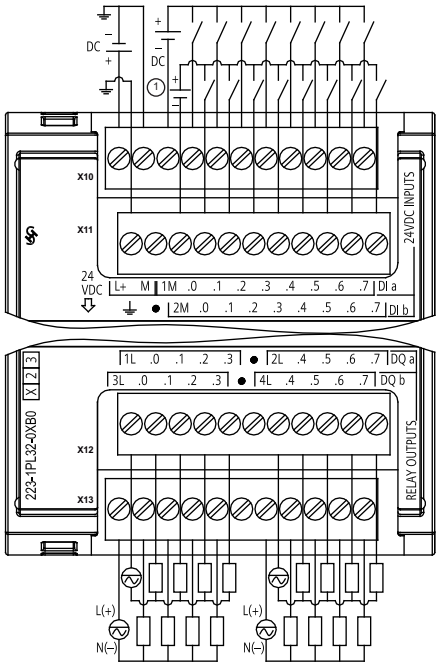
SM 1223 DI 8 x 24 VDC, DQ 8 x Relay (6ES7223-1PH32-0XB0)	**SM 1223 DI 16 x 24 VDC, DQ 16 x Relay (6ES7223-1PL32-0XB0)**	**Примечания**
		(1) Для входов-потребителей соединить "-" с "M" (см. рисунок). Для входов-источников соединить "+" с "M".

Таблица A-129 Разводка контактов для SM 1223 DI 8 x 24 VDC, DQ 8 x Relay (6ES7223-1PH32-0XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В DC	Функциональное заземление	Не подключено	Не подключено
2	M / 24 В DC	Не подключено	Не подключено	Не подключено
3	1M	2M	1L	2L
4	DI a.0	DI a.4	DQ a.0	DQ a.4
5	DI a.1	DI a.5	DQ a.1	DQ a.5
6	DI a.2	DI a.6	DQ a.2	DQ a.6
7	DI a.3	DI a.7	DQ a.3	DQ a.7

Таблица A-130 Разводка контактов для SM 1223 DI 16 x 24 VDC, DQ 16 x Relay (6ES7223-1PL32-0XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В DC	Функциональное заземление	1L	3L
2	M / 24 В DC	Не подключено	DQ a.0	DQ b.0
3	1M	2M	DQ a.1	DQ b.1

Контакт	X10	X11	X12	X13
4	DI a.0	DI b.0	DQ a.2	DQ b.2
5	DI a.1	DI b.1	DQ a.3	DQ b.3
6	DI a.2	DI b.2	Не подключено	Не подключено
7	DI a.3	DI b.3	2L	4L
8	DI a.4	DI b.4	DQ a.4	DQ b.4
9	DI a.5	DI b.5	DQ a.5	DQ b.5
10	DI a.6	DI b.6	DQ a.6	DQ b.6
11	DI a.7	DI b.7	DQ a.7	DQ b.7

Таблица A-131 Схемы электрических соединений для модулей цифрового ввода/вывода постоянного тока

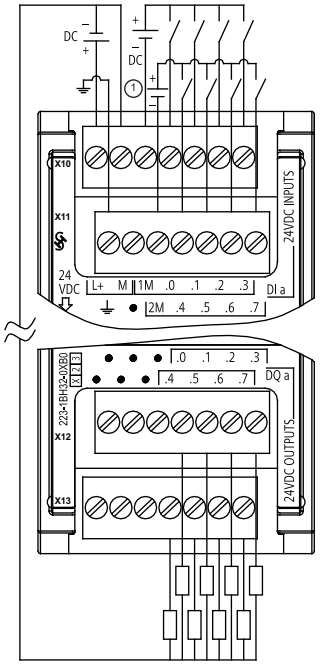
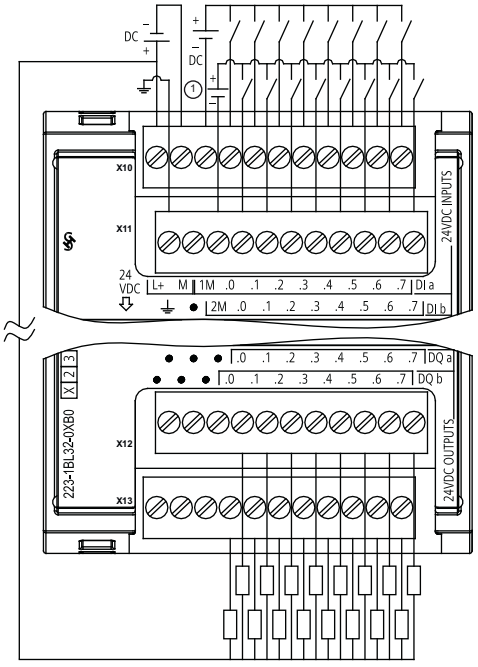
SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC (6ES7223-1BH32-0XB0)	SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC (6ES7223-1BL32-0XB0)	Примечания
		① Для входов-потребителей соединить "-" с "M" (см. рисунок). Для входов-источников соединить "+" с "M".

Таблица A-132 Разводка контактов для SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC (6ES7223-1BH32-0XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В DC	Функциональное заземление	Не подключено	Не подключено
2	M / 24 В DC	Не подключено	Не подключено	Не подключено
3	1M	2M	Не подключено	Не подключено
4	DI a.0	DI a.4	DQ a.0	DQ a.4
5	DI a.1	DI a.5	DQ a.1	DQ a.5

Контакт	X10	X11	X12	X13
6	DI a.2	DI a.6	DQ a.2	DQ a.6
7	DI a.3	DI a.7	DQ a.3	DQ a.7

Таблица A-133 Разводка контактов для SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC (6ES7223-1BL32-0XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В DC	Функциональное заземление	Не подключено	Не подключено
2	M / 24 В DC	Не подключено	Не подключено	Не подключено
3	1M	2M	Не подключено	Не подключено
4	DI a.0	DI b.0	DQ a.0	DQ b.0
5	DI a.1	DI b.1	DQ a.1	DQ b.1
6	DI a.2	DI b.2	DQ a.2	DQ b.2
7	DI a.3	DI b.3	DQ a.3	DQ b.3
8	DI a.4	DI b.4	DQ a.4	DQ b.4
9	DI a.5	DI b.5	DQ a.5	DQ b.5
10	DI a.6	DI b.6	DQ a.6	DQ b.6
11	DI a.7	DI b.7	DQ a.7	DQ b.7

Таблица A-134 Схема электрических соединений для сигнальных модулей цифрового ввода/вывода постоянного тока

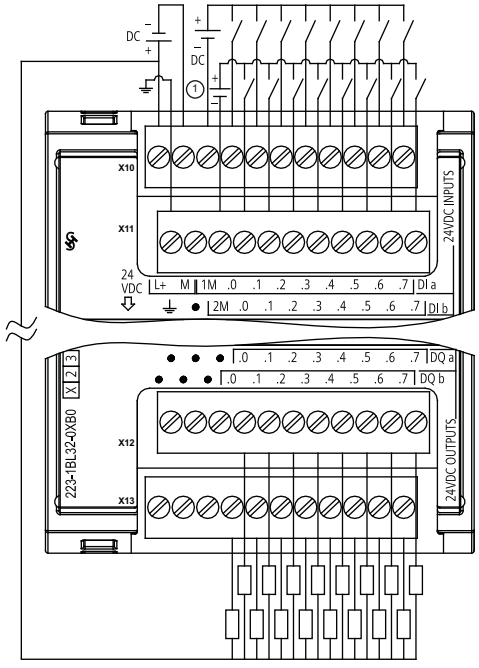
SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC потребитель (6ES7223-1BL32-1XB0)	
	

Таблица A-135 Разводка контактов для SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC потребитель (6ES7223-1BL32-1XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В DC	Функциональное заземление	Не подключено	Не подключено
2	M / 24 В DC	Не подключено	Не подключено	Не подключено
3	1M	2M	Не подключено	Не подключено
4	DI a.0	DI b.0	DQ a.0	DQ b.0
5	DI a.1	DI b.1	DQ a.1	DQ b.1
6	DI a.2	DI b.2	DQ a.2	DQ b.2
7	DI a.3	DI b.3	DQ a.3	DQ b.3
8	DI a.4	DI b.4	DQ a.4	DQ b.4
9	DI a.5	DI b.5	DQ a.5	DQ b.5
10	DI a.6	DI b.6	DQ a.6	DQ b.6
11	DI a.7	DI b.7	DQ a.7	DQ b.7

A.9.5 Технические данные для модуля цифрового ввода/вывода переменного тока SM 1223

Таблица A-136 Общие технические данные

Модель	SM 1223 DI 8 x120/230 VAC / DQ 8 x Relay
Заказной номер	6ES7223-1QH32-0XB0
Габаритные размеры (Ш x В x Г) (мм)	45 x 100 x 75
Вес	190 г
Рассеиваемая мощность	7,5 Вт
Потребляемый ток (SM шина)	120 мА
Потребляемый ток (24 В DC)	11 мА на активный выход

Таблица A-137 Цифровые входы

Модель	SM 1223 DI 8 x 120/230 VAC / DQ 8 x Relay
Количество входов	8
Тип	IEC Typ 1
Номинальное напряжение	120 В AC при 6 мА, 230 В AC при 9 мА
Длительно допустимое напряжение	264 В AC
Перенапряжение	--
Сигнал логический 1 (мин.)	79 В AC при 2,5 мА
Сигнал логический 0 (макс.)	20 В AC при 1 мА
Ток утечки (макс.)	1 мА

Модель	**SM 1223 DI 8 x 120/230 VAC / DQ 8 x Relay**
Электрическое разделение (между полевой стороной и логикой)	1500 В AC
Изолированные группы[1]	4
Время задержки ввода	Тип.: от 0,2 до 12,8 мс, выбирается пользователем макс.: -
Подключение 2-проводных датчиков приближения (Bero) (макс.)	1 мА
Длина кабеля	Неэкранированный: 300 м Экранированный: 500 м
Количество одновременно активных входов	8

[1] Каналы из одной группы должны подключаться к одной фазе.

Таблица A-138 Цифровые выходы

Модель	**SM 1223 DI 8 x 120/230 VAC / DQ 8 x Relay**
Количество входов	8
Тип	Замыкающие контакты реле
Диапазон напряжений	от 5 до 30 В DC или от 5 до 250 В AC
Сигнал логический 1 при макс. токе	--
Сигнал логический 0 при нагрузке 10 кОм	--
Ток (макс.)	2,0 А
Ламповая нагрузка	30 Вт DC/200 Вт AC
Сопротивление активного выходного канала	До 0,2 Ом для нового контакта
Ток утечки на выход	--
Импульсный ток выхода, не более	7 А через замкнутый контакт
Защита от перегрузки	Нет
Электрическое разделение (между полевой стороной и логикой)	1500 В AC (катушка - контакт) Без (катушка - логика)
Изолированные группы	2
Ток на общий провод (макс.)	10 А
Индуктивное фиксирующее напряжение	--
Задержка переключения (макс.)	10 мс
Макс. частота коммутации реле	1 Гц
Механический срок службы (без нагрузки)	10.000.000 циклов переключения
Срок службы контактов при номинальной нагрузке	100.000 циклов переключения
Поведение при переходе из RUN в STOP	Последнее значение или замещающее значение (по умолчанию 0)
Количество одновременно активных выходов	• 4 (без смежных точек) при 60 °C для горизонтальной установки или 50 °C для вертикальной установки • 8 при 55 °C для горизонтальной установки или 45 °C для вертикальной установки
Длина кабеля (в метрах)	500 м экранированный, 150 м неэкранированный

Таблица A-139 SM 1223 DI 8 x 120/230 VAC, DQ 8 x Relay (6ES7223-1QH32-0XB0)

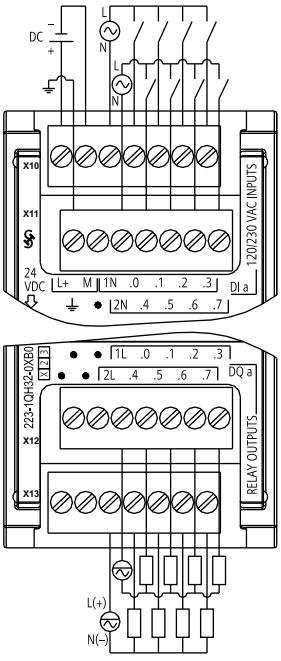


Таблица A-140 Разводка контактов для SM 1223 DI 8 x 120/240 VAC, DQ 8 x Relay (6ES7223-1QH32-0XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В DC	Функциональное заземление	Не подключено	Не подключено
2	M / 24 В DC	Не подключено	Не подключено	Не подключено
3	1N	2N	1L	2L
4	DI a.0	DI a.4	DQ a.0	DQ a.4
5	DI a.1	DI a.5	DQ a.1	DQ a.5
6	DI a.2	DI a.6	DQ a.2	DQ a.6
7	DI a.3	DI a.7	DQ a.3	DQ a.7

A.10 Аналоговые сигнальные модули (SM)

A.10.1 Технические данные модуля аналогового ввода SM 1231

Таблица A-141 Общие технические данные

Модель	**SM 1231 AI 4 x 13 бит**	**SM 1231 AI 8 x 13 бит**	**SM 1231 AI 4 x 16 бит**
Заказной номер	6ES7231-4HD32-0XB0	6ES7231-4HF32-0XB0	6ES7231-5ND32-0XB0
Габаритные размеры (Ш x В x Г) (мм)	45 x 100 x 75		
Вес	180 г		
Рассеиваемая мощность	2,2 Вт	2,3 Вт	2,0 Вт
Потребляемый ток (SM шина)	80 мА	90 мА	80 мА
Потребляемый ток (24 В DC)	45 мА		65 мА

Таблица A-142 Аналоговые входы

Модель	**SM 1231 AI 4 x 13 бит**	**SM 1231 AI 8 x 13 бит**	**SM 1231 AI 4 x 16 бит**
Количество входов	4	8	4
Тип	Напряжение или ток (дифференциальный): настраивается для каждой группы из 2 входных каналов		Напряжение или ток (дифференциальный)
Диапазон	±10 В, ±5 В, ±2,5 В, от 0 до 20 мА или от 4 до 20 мА		±10 В, ±5 В, ±2,5 В, ±1,25 В, от 0 до 20 мА или от 4 до 20 мА
Цифровое представление полной шкалы (слово данных)	от -27648 до 27648 для напряжения / от 0 до 27648 для тока		
Превышения верхней/ снижения за нижнюю границу диапазона (слово данных) См. раздел о диапазонах аналогового ввода для напряжения и тока (Страница 1548).	Напряжение: От 32511 до 27649 / от -27649 до -32512 Ток: От 32511 до 27649 / от 0 до -4864		
Переполнения за пределами верхней/нижней границы диапазона (слово данных) См. раздел о диапазонах ввода для напряжения и тока (Страница 1548).	Напряжение: От 32767 до 32512 / от -32513 до -32768 Ток от 0 до 20 мА: От 32767 до 32512 / от -4865 до -32768 Ток от 4 до 20 мА: От 32767 до 32512 (значения меньше -4864 указывают на обрыв провода)		
Разрешение1	12 бит + знаковый разряд		15 бит + знаковый разряд
Максимальное напряжение/ток	±35 В / ±40 мА		
Сглаживание	Отсутствует, слабое, среднее или сильное См. раздел Время реакции на скачок (Страница 1547)		
Подавление помех	400, 60, 50 или 10 Гц См. раздел Интервал выборки (Страница 1547)		

Модель	SM 1231 AI 4 x 13 бит	SM 1231 AI 8 x 13 бит	SM 1231 AI 4 x 16 бит
Полное входное сопротивление	≥ 9 МОм (напряжение) / ≥ 270 Ом, < 290 Ом (ток)		≥ 1 МОм (напряжение) / < 315 Ом, > 280 Ом (ток)
Электрическое разделение Между полевой стороной и логикой Между логикой и 24 В DC Между полевой стороной и 24 В DC Между каналами	Нет		 707 В DC (типовое испытание) 707 В DC (типовое испытание) 500 В DC (типовое испытание) Нет
Точность (25 °C / -20 до 60 °C)	±0,1% / ±0,2% по отношению к конечной точке шкалы		±0,1% / ±0,3% по отношению к конечной точке шкалы
Принцип измерения	Преобразование фактического значения		
Подавление синфазных сигналов	40 дБ, ном. значение при 60 Гц		
Диапазон изменения входного напряжения[1]	Суммарное значение напряжения сигнала и входного синфазного напряжения должно находиться в пределах от -12 до +12 В		
Длина кабеля (в метрах)	100 м, экранированная витая пара		

[1] Напряжения вне диапазона, подаваемые в одном канале, могут вызывать помехи в других каналах.

Таблица A-143 Диагностика

Модель	SM 1231 AI 4 x 13 бит	SM 1231 AI 8 x 13 бит	SM 1231 AI 4 x 16 бит
Переполнения за пределами верхней/нижней границы диапазона	Да		
Низкое напряжение 24 В DC	Да		
Обрыв провода	Только в диапазоне от 4 до 20 мА (если вход ниже -4864; 1,185 мА)		

Измерение тока SM 1231

Для измерения тока можно использовать 2- или 4-проводный измерительный преобразователи с подключением, представленным на следующем рисунке:

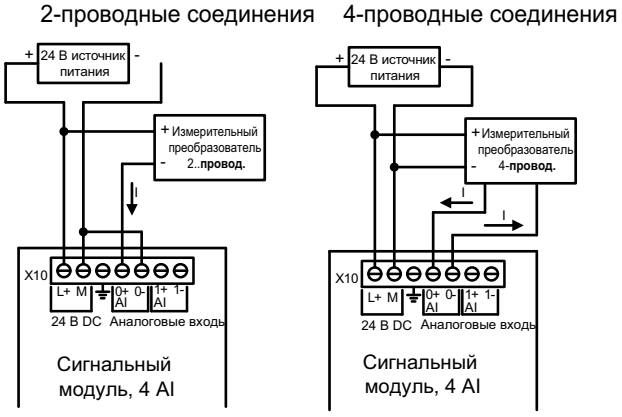


Таблица A-144 Схемы электрических соединений для SM аналогового ввода

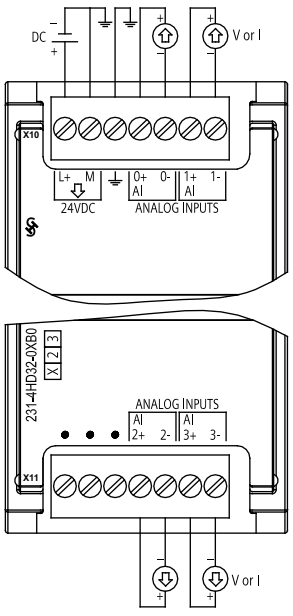
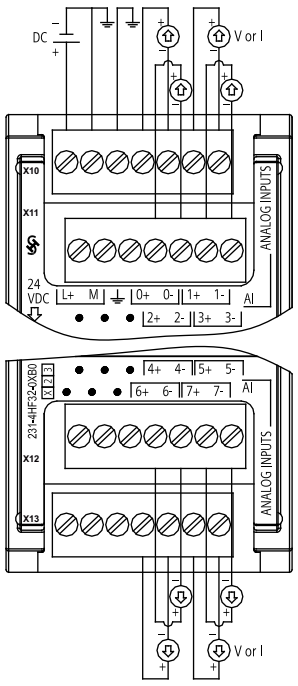
SM 1231 AI 4 x 13 бит (6ES7231-4HD32-0XB0)	**SM 1231 AI 8 x 13 бит (6ES7231-4HF32-0XB0)**
	
Примечание: Соединители должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	

Таблица A-145 Разводка контактов для SM 1231 AI 4 x 13 бит (6ES7231-4HD32-0XB0)

Контакт	X10 (позолоченный)	X11 (позолоченный)
1	L+ / 24 В DC	Не подключено
2	M / 24 В DC	Не подключено
3	Функциональное заземление	Не подключено
4	AI 0+	AI 2+
5	AI 0-	AI 2-
6	AI 1+	AI 3+
7	AI 1-	AI 3-

Таблица A-146 Разводка контактов для SM 1231 AI 8 x 13 бит (6ES7231-4HF32-0XB0)

Контакт	X10 (позолоченный)	X11 (позолоченный)	X12 (позолоченный)	X13 (позолоченный)
1	L+ / 24 В DC	Не подключено	Не подключено	Не подключено
2	M / 24 В DC	Не подключено	Не подключено	Не подключено
3	Функциональное заземление	Не подключено	Не подключено	Не подключено

Контакт	X10 (позолоченный)	X11 (позолоченный)	X12 (позолоченный)	X13 (позолоченный)
4	AI 0+	AI 2+	AI 4+	AI 6+
5	AI 0-	AI 2-	AI 4-	AI 6-
6	AI 1+	AI 3+	AI 5+	AI 7+
7	AI 1-	AI 3-	AI 5-	AI 7-

Таблица A-147 Схема электрических соединений для SM аналогового ввода

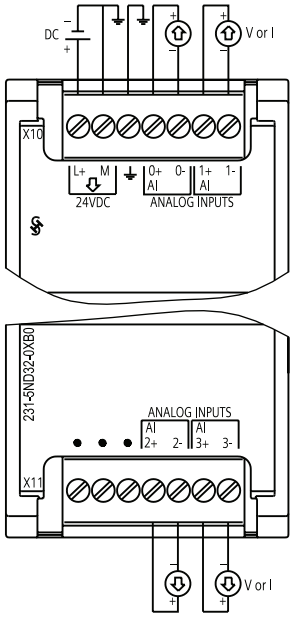
SM 1231 AI 4 x 16 бит (6ES7231-5ND32-0XB0)	
	
Примечание: Соединители должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	

Таблица A-148 Разводка контактов для SM 1231 AI 4 x 16 бит (6ES7231-5ND32-0XB0)

Контакт	X10 (позолоченный)	X11 (позолоченный)
1	L+ / 24 В DC	Не подключено
2	M / 24 В DC	Не подключено
3	Функциональное заземление	Не подключено
4	AI 0+	AI 2+
5	AI 0-	AI 2-
6	AI 1+	AI 3+
7	AI 1-	AI 3-

Примечание

Неиспользуемые входы по напряжению должны быть закорочены.

Неиспользуемые входы по току должны быть установлены на диапазон от 0 до 20 мА, и/или следует отключить сигнализацию для обрыва провода.

Входы, сконфигурированные для измерения тока, не генерируют ток в контуре, пока модуль не будет запитан и сконфигурирован.

Входы по току не будут работать, пока внешнее питание не будет подано на преобразователь.

A.10.2 Технические данные модуля аналогового вывода SM 1232

Таблица A-149 Общие технические данные

Технические данные	SM 1232 AQ 2 x 14 бит	SM 1232 AQ 4 x 14 бит
Заказной номер	6ES7232-4HB32-0XB0	6ES7232-4HD32-0XB0
Габаритные размеры (Ш x В x Г) (мм)	45 x 100 x 75	
Вес	180 г	
Рассеиваемая мощность	1,8 Вт	2,0 Вт
Потребляемый ток (SM шина)	80 мА	
Потребляемый ток (24 В DC)	45 мА (без нагрузки)	

Таблица A-150 Аналоговые выходы

Технические данные	SM 1232 AQ 2 x 14 бит	SM 1232 AQ 4 x 14 бит
Выходы	2	4
Тип	Напряжение или ток	
Диапазон	±10 В, от 0 до 20 мА или от 4 до 20 мА	
Разрешение	Напряжение: 14 бит Ток: 13 бит	
Цифровое представление полной шкалы (слово данных)	Напряжение: от -27.648 до 27.648; ток: От 0 до 27.648 См. раздел о диапазонах вывода для напряжения и тока (Страница 1549).	
Точность (25 °C / -20 до 60 °C)	0,3% / 0,6% по отношению к конечной точке шкалы	
Время установки (95% от нового значения)	Напряжение: 300 мкс (R), 750 мкс (1 мкФ) Ток: 600 мкс (1 мГн), 2 мс (10 мГн)	
Полное сопротивление нагрузки	Напряжение: ≥ 1000 Ом Ток: ≤ 600 Ом	
Максимальный выходной ток короткого замыкания	Режим по напряжению: ≤ 24 мА Режим по току: ≥ 38,5 мА	

Технические данные	SM 1232 AQ 2 x 14 бит	SM 1232 AQ 4 x 14 бит
Поведение при переходе из RUN в STOP	Последнее значение или замещающее значение (по умолчанию 0)	
Электрическое разделение (между полевой стороной и логикой)	Нет	
Электрическое разделение (между 24 В и выходом)	Нет	
Длина кабеля (в метрах)	100 м, экранированная витая пара	

Таблица A-151 Диагностика

Технические данные	SM 1232 AQ 2 x 14 бит	SM 1232 AQ 4 x 14 бит
Переполнения за пределами верхней/нижней границы диапазона	Да	
Замыкание на землю (только для режима по напряжению)	Да	
Обрыв провода (только для режима по току)[1]	Да	
24 В DC низкое напряжение[2]	Да	

[1] Обнаружение короткого замыкания возможно, только при выходном напряжении ниже -0,5 В или выше +0,5 В.

[2] Обнаружение обрыва провода возможно, только если выходной ток выше 1 мА.

Таблица A-152 Схемы электрических соединений для SM аналогового вывода

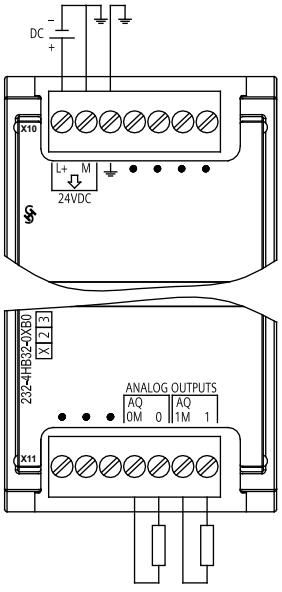
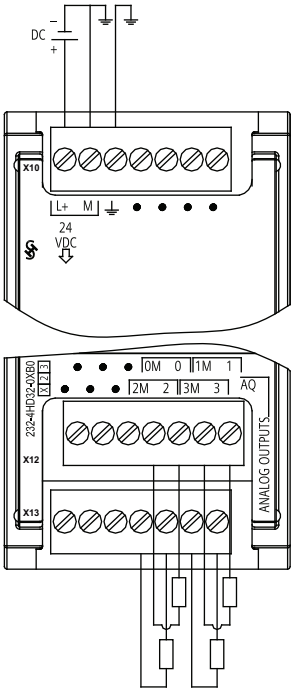
SM 1232 AQ 2 x 14 бит (6ES7232-4HB32-0XB0)	SM 1232 AQ 4 x 14 бит (6ES7232-4HD32-0XB0)
	
Примечание: Соединители должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	

Таблица A-153 Разводка контактов для SM 1232 AQ 2 x 14 бит (6ES7232-4HB32-0XB0)

Контакт	X10 (позолоченный)	X11 (позолоченный)
1	L+ / 24 В DC	Не подключено
2	M / 24 В DC	Не подключено
3	Функциональное заземление	Не подключено
4	Не подключено	AQ 0M
5	Не подключено	AQ 0
6	Не подключено	AQ 1M
7	Не подключено	AQ 1

Таблица A-154 Разводка контактов для SM 1232 AQ 4 x 14 бит (6ES7232-4HD32-0XB0)

Контакт	X10 (позолоченный)	X12 (позолоченный)	X13 (позолоченный)
1	L+ / 24 В DC	Не подключено	Не подключено
2	M / 24 В DC	Не подключено	Не подключено
3	Функциональное заземление	Не подключено	Не подключено
4	Не подключено	AQ 0M	AQ 2M
5	Не подключено	AQ 0	AQ 2
6	Не подключено	AQ 1M	AQ 3M
7	Не подключено	AQ 1	AIQ 3

A.10.3 Технические данные модуля аналогового ввода/вывода SM 1234

Таблица A-155 Общие технические данные

Технические данные	SM 1234 AI 4 x 13 бит / AQ 2 x 14 бит
Заказной номер	6ES7234-4HE32-0XB0
Габаритные размеры (Ш x В x Г) (мм)	45 x 100 x 75
Вес	220 г
Рассеиваемая мощность	2,4 Вт
Потребляемый ток (SM шина)	80 мА
Потребляемый ток (24 В DC)	60 мА (без нагрузки)

Таблица A-156 Аналоговые входы

Модель	**SM 1234 AI 4 x 13 бит / AQ 2 x 14 бит**
Количество входов	4
Тип	Напряжение или ток (дифференциальный): настраивается для каждой группы из 2 входных каналов
Диапазон	±10 В, ±5 В, ±2,5 В, от 0 до 20 мА или от 4 до 20 мА
Цифровое представление полной шкалы (слово данных)	От -27648 до 27648
Превышения верхней/снижения за нижнюю границу диапазона (слово данных)	Напряжение: От 32511 до 27649 / от -27649 до -32512 Ток: От 32511 до 27649 / от 0 до -4864 См. раздел о диапазонах ввода для напряжения и тока (Страница 1548).
Переполнения за пределами верхней/нижней границы диапазона (слово данных)	Напряжение: От 32767 до 32512 / от -32513 до -32768 Ток: От 32767 до 32512 / от -4865 до -32768 См. раздел о диапазонах ввода для напряжения и тока (Страница 1548).
Разрешение	12 бит + знаковый разряд
Максимальное напряжение/ток	±35 В / ±40 мА
Сглаживание	Отсутствует, слабое, среднее или сильное См. раздел Время реакции на скачок (Страница 1547)
Подавление помех	400, 60, 50 или 10 Гц См. раздел Интервал выборки (Страница 1547)
Полное входное сопротивление	≥ 9 МОм (напряжение) / ≥ 270 Ом, < 290 Ом (ток)
Электрическое разделение (между полевой стороной и логикой)	Нет
Точность (25 °C / -20 до 60 °C)	±0,1% / ±0,2% по отношению к конечной точке шкалы
Время преобразования аналогового значения в цифровое значение	625 мкс (подавление помех 400 Гц)
Подавление синфазных сигналов	40 дБ, ном. значение при 60 Гц
Диапазон изменения входного напряжения[1]	Суммарное значение напряжения сигнала и входного синфазного напряжения должно находиться в пределах от -12 до +12 В
Длина кабеля (в метрах)	100 м, экранированная витая пара

[1] Напряжения вне диапазона, подаваемые в одном канале, могут вызывать помехи в других каналах.

Таблица A-157 Аналоговые выходы

Технические данные	**SM 1234 AI 4 x 13 бит / AQ 2 x 14 бит**
Выходы	2
Тип	Напряжение или ток
Диапазон	±10 В или от 0 до 20 мА или от 4 до 20 мА
Разрешение	Напряжение: 14 битов, ток: 13 бит
Цифровое представление полной шкалы (слово данных)	Напряжение: от -27.648 до 27.648; ток: от 0 до 27648 См. раздел о диапазонах вывода для напряжения и тока (Страница 1549).
Точность (25 °C / -20 до 60 °C)	0,3% / 0,6% по отношению к конечной точке шкалы

Технические данные	SM 1234 AI 4 x 13 бит / AQ 2 x 14 бит
Время установки (95% от нового значения)	Напряжение: 300 мкс (R), 750 мкс (1 мкФ) Ток: 600 мкс (1 мГн), 2 мс (10 мГн)
Полное сопротивление нагрузки	Напряжение: ≥ 1000 Ом Ток: ≤ 600 Ом
Максимальный выходной ток короткого замыкания	Режим по напряжению: ≤ 24 мА Режим по току: ≥ 38,5 мА
Поведение при переходе из RUN в STOP	Последнее значение или замещающее значение (по умолчанию 0)
Электрическое разделение (между полевой стороной и логикой)	Нет
Электрическое разделение (между 24 В и выходом)	Нет
Длина кабеля (в метрах)	100 м, экранированная витая пара

Таблица A-158 Диагностика

Модель	SM 1234 AI 4 x 13 бит / AQ 2 x 14 бит
Переполнения за пределами верхней/нижней границы диапазона	Да
Замыкание на землю (только для режима по напряжению)[1]	Да, на выходах
Обрыв провода (только для режима по току)[2]	Да, на выходах
Низкое напряжение 24 В DC	Да

[1] Обнаружение короткого замыкания возможно, только при выходном напряжении ниже -0,5 В или выше +0,5 В.

[2] Обнаружение обрыва провода возможно, только если выходной ток выше 1 мА.

Измерение тока SM 1234

Для измерения тока можно использовать 2- или 4-проводный измерительный преобразователи с подключением, представленным на следующем рисунке:

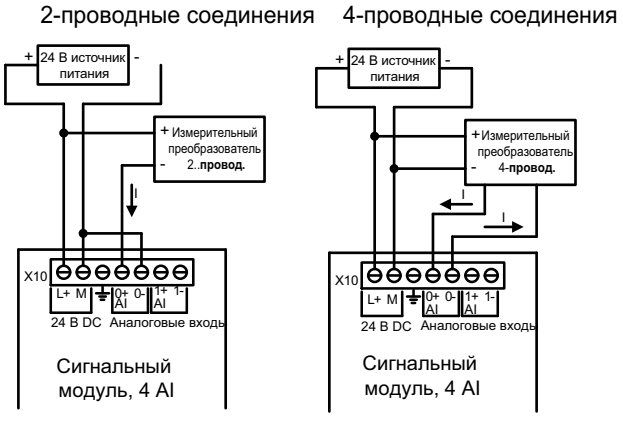


Таблица A-159 Схемы электрических соединений SM аналогового ввода/вывода

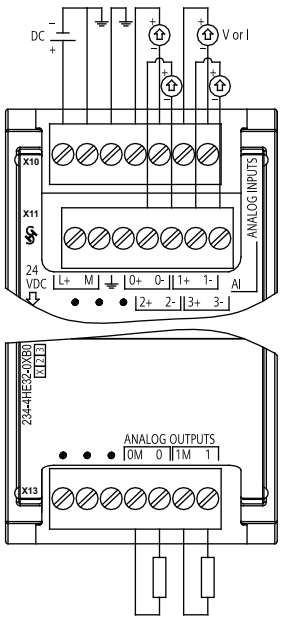
SM 1234 AI 4 x 13 бит / AQ 2 x 14 бит (6ES7234-4HE32-0XB0)	
	
Примечание: Соединители должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	

Таблица A-160 Разводка контактов для SM 1234 AI 4 x 13 бит / AQ 2 x 14 бит (6ES7234-4HE32-0XB0)

Контакт	X10 (позолоченный)	X11 (позолоченный)	X13 (позолоченный)
1	L+ / 24 В DC	Не подключено	Не подключено
2	M / 24 В DC	Не подключено	Не подключено
3	Функциональное заземление	Не подключено	Не подключено
4	AI 0+	AI 2+	AQ 0M
5	AI 0-	AI 2-	AQ 0
6	AI 1+	AI 3+	AQ 1M
7	AI 1-	AI 3-	AQ 1

Примечание

Неиспользуемые входы по напряжению должны быть закорочены.

Неиспользуемые входы по току должны быть установлены на диапазон от 0 до 20 мА, и/или следует отключить сигнализацию для обрыва провода.

Входы, сконфигурированные для измерения тока, не генерируют ток в контуре, пока модуль не будет запитан и сконфигурирован.

Входы по току не будут работать, пока внешнее питание не будет подано на преобразователь.

A.10.4 Реакция на скачок аналоговых входов

Таблица A-161 Реакция на скачок (мс), от 0 до границы диапазона, измеренная при 95%

Выбор сглаживания (усреднение по выборке)	**Частота подавления помех (выбор времени интегрирования)**			
	400 Гц (2,5 мс)	**60 Гц (16,6 мс)**	**50 Гц (20 мс)**	**10 Гц (100 мс)**
Отсутствует (1 цикл): без усреднения	4 мс	18 мс	22 мс	100 мс
Слабое (4 цикла): 4 значения	9 мс	52 мс	63 мс	320 мс
Среднее (16 циклов): 16 значений	32 мс	203 мс	241 мс	1200 мс
Сильное (32 цикла): 32 значения	61 мс	400 мс	483 мс	2410 мс

A.10.5 Время выборки и время обновления для аналоговых входов

Таблица A-162 Время выборки и обновления модуля для всех каналов

Частота подавления помех (время интегрирования)	**Время выборки и обновления модуля для всех каналов**			
	400 Гц (2,5 мс)	**60 Гц (16,6 мс)**	**50 Гц (20 мс)**	**10 Гц (100 мс)**
SM 4-канальный x 13 бит	0,625 мс	4,17 мс	5 мс	25 мс
SM 8-канальный x 13 бит	1,25 мс	4,17 мс	5 мс	25 мс
SM 4-канальный x 16 бит	0,417 мс	0,397 мс	0,400 мс	0,400 мс

A.10.6 Диапазоны измерений аналоговых входов для напряжения и тока (SB и SM)

Таблица A-163 Представление аналогового входа для напряжения (SB и SM)

Система		**Диапазон измерения напряжения**				
Десятичная	**Шестнадцатеричная**	**±10 В**	**±5 В**	**±2,5 В**	**±1,25 В**	
32767	7FFF[1]	11,851 В	5,926 В	2,963 В	1,481 В	Выход за верхний установленный предел
32512	7F00					
32511	7EFF	11,759 В	5,879 В	2,940 В	1,470 В	Выход за границы диапазона
27649	6C01					
27648	6C00	10 В	5 В	2,5 В	1,250 В	Расчетный диапазон
20736	5100	7,5 В	3,75 В	1,875 В	0,938 В	
1	1	361,7 мкВ	180,8 мкВ	90,4 мкВ	45,2 мкВ	
0	0	0 В	0 В	0 В	0 В	
-1	FFFF					
-20736	AF00	-7,5 В	-3,75 В	-1,875 В	-0,938 В	
-27648	9400	-10 В	-5 В	-2,5 В	-1,250 В	
-27649	93FF					Выход за нижнюю границы диапазона
-32512	8100	-11,759 В	-5,879 В	-2,940 В	-1,470 В	
-32513	80FF					Выход сигнала за нижний установленный предел
-32768	8000	-11,851 В	-5,926 В	-2,963 В	-1,481 В	

[1] Возврат 7FFF возможен по одной из следующих причин: Выход за верхний установленный предел (см. таблицу), до получения правильных значений (например, сразу же при запуске) или при обнаружении обрыва провода.

Таблица A-164 Представление аналогового входа для тока (SB и SM)

Система		**Диапазон измерения тока**		
Десятичная	**Шестнадцатеричная**	**от 0 до 20 мА**	**от 4 до 20 мА**	
32767	7FFF	> 23,52 мА	> 22,81 мА	Выход за верхний установленный предел
32511	7EFF	23,52 мА	22,81 мА	Выход за границы диапазона
27649	6C01			
27648	6C00	20 мА	20 мА	Ном. диапазон
20736	5100	15 мА	16 мА	
1	1	723,4 нА	4 мА + 578,7 нА	
0	0	0 мА	4 мА	
-1	FFFF			Выход за нижнюю границы диапазона
-4864	ED00	-3,52 мА	1,185 мА	

Система		Диапазон измерения тока		
Десятичная	**Шестнадцатеричная**	**от 0 до 20 мА**	**от 4 до 20 мА**	
32767[1]	7FFF		< - 1,185 мА	Обрыв провода (от 4 до 20 мА)
-32768	8000	< -3,52 мА		Выход сигнала за нижний установленный предел (от 0 до 20 мА)

[1] Значение обрыва провода 32767 (16#7FFF) возвращается всегда, независимо от состояния предупреждения об обрыве провода.

См. также

Определение типа обрыва провода с помощью модуля SM 1231 (Страница 1423)

A.10.7 Диапазоны измерений аналоговых выходов для напряжения и тока (SB и SM)

Таблица A-165 Представление аналогового выхода для напряжения (SB и SM)

Система		Диапазон выходного напряжения	
Десятичная	**Шестнадцатеричная**	**±10 В**	
32767	7FFF	См. Примечание 1	Выход за верхний установленный предел
32512	7F00	См. Примечание 1	
32511	7EFF	11,76 В	Выход за границы диапазона
27649	6C01		
27648	6C00	10 В	Расчетный диапазон
20736	5100	7,5 В	
1	1	361,7 мкВ	
0	0	0 В	
-1	FFFF	-361,7 мкВ	
-20736	AF00	-7,5 В	
-27648	9400	-10 В	
-27649	93FF		Выход за нижнюю границы диапазона
-32512	8100	-11,76 В	
-32513	80FF	См. Примечание 1	Выход сигнала за нижний установленный предел
-32768	8000	См. Примечание 1	

[1] При выходе за верхний или нижний установленный предел, аналоговые выходы используют замещающее значение для рабочего состояния STOP.

Таблица A-166 Представление аналогового выхода для тока (SB и SM)

Система		Диапазон выходного тока		
Десятичная	**Шестнадца-теричная**	**от 0 до 20 мА**	**от 4 до 20 мА**	
32767	7FFF	См. Примечание 1	См. Примечание 1	Выход за верхний установленный предел
32512	7F00	См. Примечание 1	См. Примечание 1	
32511	7EFF	23,52 мА	22,81 мА	Выход за границы диапазона
27649	6C01			
27648	6C00	20 мА	20 мА	Расчетный диапазон
20736	5100	15 мА	16 мА	
1	1	723,4 нА	4 мА + 578,7 нА	
0	0	0 мА	4 мА	
-1	FFFF		4 мА до 578,7 нА	Выход за нижнюю границы диапазона
-6912	E500		0 мА	
-6913	E4FF			Невозможно. Значение выходного сигнала ограничено до 0 мА.
-32512	8100			
-32513	80FF	См. Примечание 1	См. Примечание 1	Выход сигнала за нижний установленный предел
-32768	8000	См. Примечание 1	См. Примечание 1	

[1] При выходе за верхний или нижний установленный предел, аналоговые выходы используют замещающее значение для рабочего состояния STOP.

A.11 Сигнальные модули для термопар и термосопротивлений (SM)

A.11.1 SM 1231 Thermocouple (TC)

Таблица A-167 Общие технические данные

Модель	**SM 1231 AI 4 x 16 бит TC**	**SM 1231 AI 8 x 16 бит TC**
Заказной номер	6ES7231-5QD32-0XB0	6ES7231-5QF32-0XB0
Габаритные размеры (Ш x В x Г) (мм)	45 x 100 x 75	
Вес	180 г	190 г
Рассеиваемая мощность	1,5 Вт	
Потребляемый ток (SM шина)	80 мА	
Потребляемый ток (24 В DC) [1]	40 мА	

[1] От 20,4 до 28,8 В DC (класс 2, ограниченное питание или питание датчиков от PLC)

Таблица A-168 Аналоговые входы

Модель		**SM 1231 AI 4 x 16 бит TC**	**SM 1231 AI 8 x 16 бит TC**
Количество входов		4	8
Диапазон Ном. диапазон (слово данных) Переполнения за пределами верхней/нижней границы диапазона (слово данных) Переполнения за пределами верхней/нижней границы диапазона (слово данных)		См. Таблицу выбора термопар (Страница 1554).	
Разрешение	Температура	0,1 °C / 0,1 °F	
	Напряжение	15 бит плюс знаковый разряд	
Максимальное входное напряжение		±35 В	
Подавление шумов		85 дБ для выбранной настройки фильтра (10 Гц, 50 Гц, 60 Гц или 400 Гц)	
Подавление синфазных сигналов		>120 дБ при 120 В AC	
Полное сопротивление		≥ 10 МОм	
Электрическое разделение	Между полевой стороной и логикой	707 В DC (типовое испытание)	
	Между полевой стороной и 24 В DC	707 В DC (типовое испытание)	
	Между 24 В DC и логикой	707 В DC (типовое испытание)	
Между каналами		120 В AC	
Точность		См. Таблицу выбора термопар (Страница 1554).	
Повторяемость		±0,05 % FS	
Принцип измерения		Интегрирование	
Время обновления модуля		См. таблицу выбора для способа подавления помех (Страница 1554).	
Ошибка холодного спая		±1,5 В	
Длина кабеля (в метрах)		Макс. 100 м до датчика	
Сопротивление линии		Макс. 100 Ом	

Таблица A-169 Диагностика

Модель	**SM 1231 AI 4 x 16 бит TC**	**SM 1231 AI 8 x 16 бит TC**
Переполнения за пределами верхней/нижней границы диапазона[1]	Да	
Обрыв провода[2, 3]	Да	
24 В DC низкое напряжение [1]	Да	

[1] Диагностическая информация о переполнении и низком напряжении будет передана в аналоговых значениях, даже если тревоги будут отключены в конфигурации модуля.

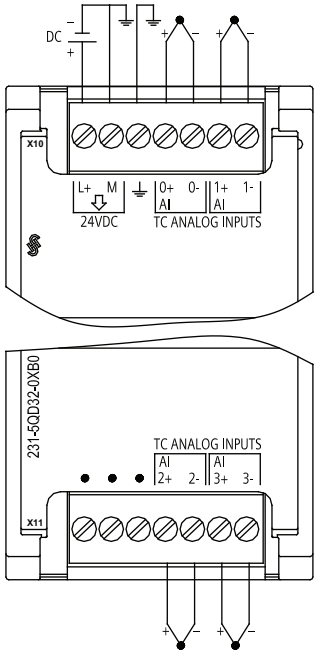
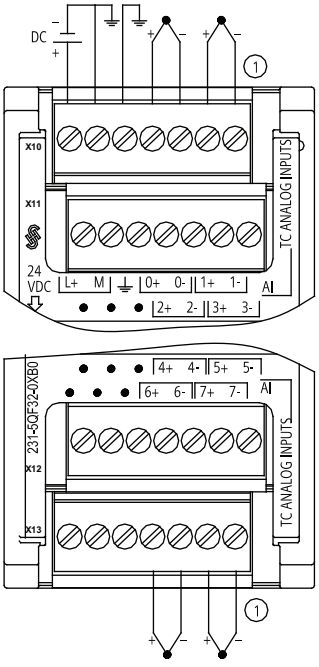
[2] Если тревога обрыва провода отключена, и имеет место обрыв в проводке датчика, модуль может возвращать случайные значения.

[3] Модуль выполняет контроль обрыва провода каждые 6 секунд, из-за чего время обновления каждые 6 секунд увеличивается на 9 мс для каждого активированного канала.

Аналоговый сигнальный модуль SM 1231 Thermocouple (TC) измеряет значение напряжения, поданного на входы модуля. Типом измерения температуры может быть "Термопара" или "Напряжение".

- "Термопара": Измеренное значение будет передано в градусах, умноженных на десять (например: 25,3 градуса будет представлено как десятичное число 253).
- "Напряжение": Номинальным предельным значением диапазона измерения будет десятичное 27648.

Таблица A-170 Схемы электрических соединений модуля SM Thermocouple (TC)

SM 1231 AI 4 x TC 16 бит (6ES7231-5QD32-0XB0)	**SM 1231 AI 8 x TC бит (6ES7231-5QF32-0XB0)**
	
Примечание: Соединители должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	

① Соединения TC 2, 3, 4 и 5 не показаны для лучшей наглядности.

Таблица A-171 Разводка контактов для SM 1231 AI 4 x TC 16 бит (6ES7231-5QD32-0XB0)

Контакт	X10 (позолоченный)	X11 (позолоченный)
1	L+ / 24 В DC	Не подключено
2	M / 24 В DC	Не подключено
3	Функциональное заземление	Не подключено
4	AI 0+ /TC	AI 2+ /TC
5	AI 0- /TC	AI 2- /TC
6	AI 1+ /TC	AI 3+ /TC
7	AI 1- /TC	AI 3- /TC

Таблица A-172 Разводка контактов для SM 1231 AI 8 x TC бит (6ES7231-5QF32-0XB0)

Контакт	X10 (позолоченный)	X11 (позолоченный)	X12 (позолоченный)	X13 (позолоченный)
1	L+ / 24 В DC	Не подключено	Не подключено	Не подключено
2	M / 24 В DC	Не подключено	Не подключено	Не подключено
3	Функциональное заземление	Не подключено	Не подключено	Не подключено
4	AI 0+ /TC	AI 2+ /TC	AI 4+ /TC	AI 6+ /TC
5	AI 0- /TC	AI 2- /TC	AI 4- /TC	AI 6- /TC
6	AI 1+ /TC	AI 3+ /TC	AI 5+ /TC	AI 7+ /TC
7	AI 1- /TC	AI 3- /TC	AI 5- /TC	AI 7- /TC

Примечание

Неиспользуемые аналоговые входы должны быть закорочены.

Неиспользуемые каналы для термопары могут быть отключены. Отключение неиспользуемых каналов не приводит к ошибкам.

A.11.1.1 Принцип работы термопары

Термопары образуются, когда два разнородных металла электрически соединяются друг с другом. При соединении вырабатывается напряжение, которое пропорционально температуре в месте спая. Это напряжение является очень низким. Один микровольт может соответствовать нескольким градусам. Измерение напряжения термопары, компенсация дополнительных спаев и последующая линеаризация результатов являются основанием для измерения температуры с помощью термопар.

При подключении термопары к модулю SM 1231 Thermocouple, оба разнородных металлических провода подключаются к сигнальному соединению модуля. Место, где два разнородных провода соединены друг с другом, образует термопару датчика.

Еще две термопары образуются, когда два разнородных провода подключаются к сигнальному соединителю. Температура соединителя создает напряжение, которое добавляется к напряжению от термопары датчика. Если это напряжение не скорректировано, то передаваемая температура, будет отличаться от температуры датчика.

Компенсация холодного спая используется для корректировки влияния термопары соединителя. Таблицы для термопар основаны на опорной температуре соединения, обычно это нуль градусов по Цельсию. Компенсация холодного спая дает корректировку соединителю на ноль градусов по Цельсию. Компенсация холодного спая исправляет напряжение, измененное термопарой соединителя. Температура модуля измеряется внутри, и затем преобразуется в значение, которое будет добавлено к значению от датчика. Затем исправленное значение датчика линеаризуется, используя таблицы термопар.

Для оптимальной работы компенсации холодного спая модуль термопары должен быть расположен в среде со стабильной температурой. Медленное изменение (менее чем 0,1 °C/мин) температуры в месте нахождения модуля правильно компенсируется согласно

техническим требованиям модуля. Движение воздуха через модуль также приводит к ошибкам компенсации холодного спая.

Для наилучшей компенсации ошибки холодного спая, можно использовать внешнюю изотермическую клеммную колодку. Модуль термопары поддерживает использование клеммной колодки с эталонной температурой 0 °C или 50 °C.

A.11.1.2 Таблицы выбора для SM 1231 Thermocouple (TC)

Диапазоны и точность для различных типов термопар, поддерживаемых сигнальным модулем SM 1231 Thermocouple, представлены в таблице ниже.

Таблица A-173 Таблицу выбора термопар

Тип	Минимальная нижняя граница температуры[1]	Номинальная нижняя граница температуры	Номинальная верхняя граница температуры	Максимальная верхняя граница температуры[2]	Погрешность преобразования [3, 4] при 25 °C	Погрешность преобразования [1, 2] при -20 °C до 60 °C
J	-210,0 °C	-150,0 °C	1200,0 °C	1450,0 °C	±0,3 °C	±0,6 °C
	-346,0 °F	-238,0 °F	2192,0 °F	2642,0 °F	±0,5°F	±1,1 °F
K	-270,0 °C	-200,0 °C	1372,0 °C	1622,0 °C	±0,4 °C	±1,0 °C
	-454,0 °F	-328,0 °F	2501,6 °F	2951,6 °F	±0,7 °F	±1,8 °F
T	-270,0 °C	-200,0 °C	400,0 °C	540,0 °C	±0,5 °C	±1,0 °C
	-454,0 °F	-328,0 °F	752,0 °F	1004,0 °F	±0,9 °F	±1,8 °F
E	-270,0 °C	-200,0 °C	1000,0 °C	1200,0 °C	±0,3 °C	±0,6 °C
	-454,0 °F	-328,0 °F	1832,0 °F	2192,0 °F	±0,5°F	±1,1 °F
R & S	-50,0 °C	100,0 °C	1768,0 °C	2019,0 °C	±1,0 °C	±2,5 °C
	-58,0 °C	212,0 °F	3214,4 °F	3276,6 °F [5]	±1,8 °F	±4,5 °F
B	0,0 °C	200,0 °C	800,0 °C	--	±2,0 °C	±2,5 °C
	32,0 °F	392,0 °F	1472,0 °F	--	±3,6 °F	±4,5 °F
	--	800,0 °C	1820,0 °C	1820,0 °C	±1,0 °C	±2,3 °C
	--	1472,0 °F	3276,6 °F [5]	3276,6 °F [5]	±1,8 °F	±4,1 °F
N	-270,0 °C	-200,0 °C	1300,0 °C	1550,0 °C	±1,0 °C	±1,6 °C
	-454,0 °F	-328,0 °F	2372,0 °F	2822,0 °F	±1,8 °F	±2,9 °F
C	0,0 °C	100,0 °C	2315,0 °C	2500,0 °C	±0,7 °C	±2,7 °C
	32,0 °F	212,0 °F	3276,6 °F [5]	3276,6 °F [5]	±1,3 °F	±4,9 °F

Тип	Минимальная нижняя граница температуры[1]	Номинальная нижняя граница температуры	Номинальная верхняя граница температуры	Максимальная верхняя граница температуры[2]	Погрешность преобразования [3, 4] при 25 °C	Погрешность преобразования [1, 2] при -20 °C до 60 °C
TXK/XK(L)	-200,0 °C	-150,0 °C	800,0 °C	1050,0 °C	±0,6 °C	±1,2 °C
	-328,0 °F	302,0 °F	1472,0 °F	1922,0 °F	±1,1 °F	±2,2 °F
Напряжение	-32512	-27648 -80 мВ	27648 80 мВ	32511	±0,05 %	±0,1 %

1 Значения от термопары ниже минимальной нижней границы температуры возвращаются как -32768.

2 Значения от термопары выше максимальной верхней границы температуры возвращаются как 32767.

3 Погрешность внутренней температурной компенсации холодного спая составляет ±1,5% для всех диапазонов измерения температуры. Эта погрешность добавляется к приведенным в таблице погрешностям. Для выхода на эти показатели модуль должен отработать не менее 30 минут.

4 При наличии помех в диапазоне частот от 970 до 990 МГц точность преобразования SM 1231 AI 4 x 16 бит TC может снижаться.

5 Нижнее предельное значение 3276,6 с информацией в °F

Примечание

Канал для подключения термопары

Каждый из каналов на сигнальном модуле Thermocouple может конфигурироваться для различных типов термопар (можно выбрать в программном обеспечении во время конфигурации модуля).

Таблица A-174 Подавление помех и время обновления для SM 1231 Thermocouple (TC)

Выбор частоты подавления помех	Время интегрирования	Время обновления данных для 4-канального модуля (сек)	Время обновления данных для 8-канального модуля (сек)
400 Гц (2,5 мс)	10 мс[1]	0,143	0,285
60 Гц (16,6 мс)	16,67 мс	0,223	0,445
50 Гц (20 мс)	20 мс	0,263	0,525
10 Гц (100 мс)	100 мс	1,225	2,450

1 Для того, чтобы достичь заявленной точности и разрешения для модуля при фильтре 400 Гц, время интегрирования составляет 10 мс. Этот фильтр также отсекает помехи частот 100 Гц и 200 Гц.

Для измерительных термопар рекомендуется использовать время интегрирования в 100 мс. Использование более низких значений времени интегрирования увеличивает погрешность повторяемости при измерении температуры.

Примечание

После включения модуль выполняет внутреннюю калибровку аналого-цифрового преобразователя. В течение этого времени модуль возвращает значение 32767 в каждом канале, пока действительные данные не станут доступны на этом канале. При необходимости следует учесть это время инициализации в программе пользователя. Поскольку конфигурация модуля может изменить продолжительность времени инициализации, следует проверить поведение модуля в конкретной конфигурации. При необходимости можно включать логику в свою программу пользователя, чтобы подстроиться под время инициализации модуля.

Представление аналоговых значений для термопары типа J

Аналоговые значения для термопары типа J представлены в таблице ниже.

Таблица A-175 Представление аналоговых значений для термопары типа J

Тип J в °C	Единицы		Тип J в °F	Единицы		Диапазон
	Десятичные	Шестнадцатеричные		Десятичные	Шестнадцатеричные	
> 1450,0	32767	7FFF	> 2642,0	32767	7FFF	Выход за верхний установленный предел
1450,0 : 1200,1	14500 : 12001	38A4 : 2EE1	2642,0 : 2192,2	26420 : 21922	6734 : 55A2	Выше диапазона
1200,0 : -150,0	12000 : -1500	2EE0 : FA24	2192,0 : -238,0	21920 : -2380	55A0 : F6B4	Расчетный диапазон
< -150,0	-32768	8000	< -238,0	-32768	8000	Выход за нижний установленный предел[1]

[1] Неправильное подключение (к примеру, обратная полярность или разомкнутые входы) или ошибка датчика в диапазоне отрицательных температур (к примеру, неверный тип термопары) может стать причиной выхода за нижний установленный предел в сигнальном модуле для подключения термопар.

A.11.2 SM 1231 RTD

Технические данные SM 1231 RTD

Таблица A-176 Общие технические данные

Технические данные	SM 1231 AI 4 x RTD x 16 бит	SM 1231 AI 8 x RTD x 16 бит
Заказной номер	6ES7231-5PD32-0XB0	6ES7231-5PF32-0XB0
Габаритные размеры (Ш x В x Г) (мм)	45 x 100 x 75	70 x 100 x 75
Вес	220 г	270 г
Рассеиваемая мощность	1,5 Вт	
Потребляемый ток (SM шина)	80 мА	90 мА
Потребляемый ток (24 В DC) [1]	40 мА	

[1] От 20,4 до 28,8 В DC (класс 2, ограниченное питание или питание датчиков от CPU)

Таблица A-177 Аналоговые входы

Технические данные		SM 1231 AI 4 x RTD x 16 бит	SM 1231 AI 8 x RTD x16 бит
Количество входов		4	8
Тип		Рекомендованные для модуля термометры сопротивления и сопротивления	
Диапазон Ном. диапазон (слово данных) Превышения верхней/снижения за нижнюю границу диапазона (слово данных) Переполнения за пределами верхней/нижней границы диапазона (слово данных)		См. таблицу выбора для термометров сопротивления (Страница 1560).	
Разрешение	Температура	0,1 °C / 0,1 °F	
	Сопротивление	15 бит плюс знаковый разряд	
Максимальное входное напряжение		±35 В	
Подавление шумов		85 дБ для режимов фильтрации с частотами 10, 50, 60 или 400 Гц	
Подавление синфазных сигналов		>120 дБ	
Полное сопротивление		≥ 10 МОм	
Электрическое разделение	Между полевой стороной и логикой	707 В DC (типовое испытание)	
	Между полевой стороной и 24 В DC	707 В DC (типовое испытание)	
	Между 24 В DC и логикой	707 В DC (типовое испытание)	
Развязка между каналами		Нет	
Точность		См. таблицу выбора для термометров сопротивления (Страница 1560).	

Технические данные	SM 1231 AI 4 x RTD x 16 бит	SM 1231 AI 8 x RTD x16 бит
Повторяемость	±0,05 % FS	
Потери мощности датчика, не более	0,5 мВт	
Принцип измерения	Интегрирование	
Время обновления модуля	См. таблицу выбора для способа подавления помех (Страница 1560).	
Длина кабеля (в метрах)	Макс. 100 м до датчика	
Сопротивление линии	20 Ом, 2,7 Ом для 10 Ом термометров сопротивления	

Таблица A-178 Диагностика

Технические данные	SM 1231 AI 4 x RTD x 16 бит	SM 1231 AI 8 x RTD x16 бит
Переполнения за пределами верхней/нижней границы диапазона[1,2]	Да	
Обрыв провода[3]	Да	
24 В DC низкое напряжение [1]	Да	

[1] Диагностическая информация о переполнении и низком напряжении будет передана в аналоговых значениях, даже если тревоги будут отключены в конфигурации модуля.

[2] Для диапазонов измерения сопротивления проверка на переполнения за пределами нижней границы диапазона никогда не активируется.

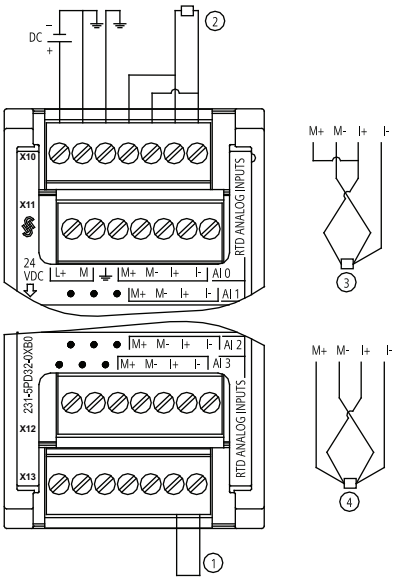
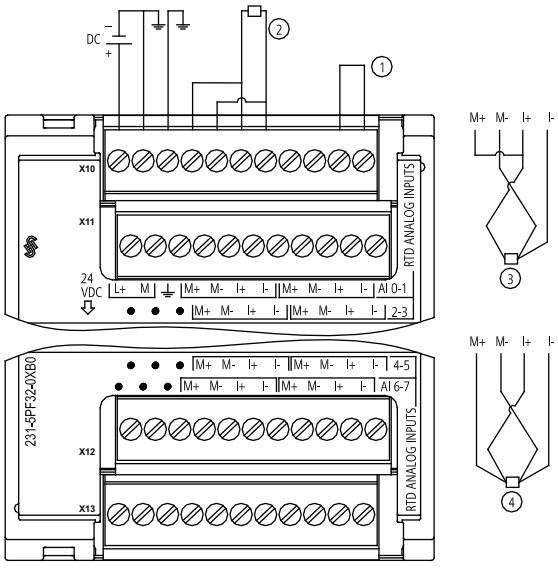
[3] Если тревога обрыва провода отключена, и имеет место обрыв в проводке датчика, модуль может возвращать случайные значения.

Аналоговый сигнальный модуль SM 1231 RTD измеряет значение сопротивления, подключенного ко входам модуля. В качестве тип измерения можно выбрать "Сопротивление" или "Тепловое сопротивление".

- "Сопротивление": Номинальным предельным значением диапазона измерения будет десятичное 27648.
- "Тепловое сопротивление": Измеренное значение будет передано в градусах, умноженных на десять (например: 25,3 градуса будет представлено как десятичное число 253). Значения климатического диапазона будут переданы в градусах, умноженных на сто (например: 25,34 градуса будет представлено как десятичное число 2534).

Сигнальный модуль SM 1231 RTD поддерживает измерения с помощью 2-проводного, 3-проводного и 4-проводного подключения датчика.

Таблица A-179 Схемы электрических соединений RTD-SM

SM 1231 RTD 4 x 16 бит (6ES7231-5PD32-0XB0)	SM 1231 RTD 8 x 16 бит (6ES7231-5PF32-0XB0)
	

① Закольцевать не занятые входы RTD

② 2-проводный RTD ③ 3-проводный RTD ④ 4-проводный RTD

ПРИМЕЧАНИЕ: Соединители должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.

Таблица A-180 Разводка контактов для SM SM 1231 RTD 4 x 16 бит (6ES7231-5PD32-0XB0)

Контакт	X10 (позолоченный)	X11 (позолоченный)	X12 (позолоченный)	X13 (позолоченный)
1	L+ / 24 В DC	Не подключено	Не подключено	Не подключено
2	M / 24 В DC	Не подключено	Не подключено	Не подключено
3	Функциональное заземление	Не подключено	Не подключено	Не подключено
4	AI 0 M+ /RTD	AI 1 M+ /RTD	AI 2 M+ /RTD	AI 3 M+ /RTD
5	AI 0 M- /RTD	AI 1 M- /RTD	AI 2 M- /RTD	AI 3 M- /RTD
6	AI 0 I+ /RTD	AI 1 I+ /RTD	AI 2 I+ /RTD	AI 3 I+ /RTD
7	AI 0 I- /RTD	AI 1 I- /RTD	AI 2 I- /RTD	AI 3 I- /RTD

Таблица A-181 Разводка контактов для SM 1231 RTD 8 x 16 бит (6ES7231-5PF32-0XB0)

Контакт	X10 (позолоченный)	X11 (позолоченный)	X12 (позолоченный)	X13 (позолоченный)
1	L+ / 24 В DC	Не подключено	Не подключено	Не подключено
2	M / 24 В DC	Не подключено	Не подключено	Не подключено

Контакт	X10 (позолоченный)	X11 (позолоченный)	X12 (позолоченный)	X13 (позолоченный)
3	Функциональное заземление	Не подключено	Не подключено	Не подключено
4	AI 0 M+ /RTD	AI 2 M+ /RTD	AI 4 M+ /RTD	AI 6 M+ /RTD
5	AI 0 M- /RTD	AI 2 M- /RTD	AI 4 M- /RTD	AI 6 M- /RTD
6	AI 0 I+ /RTD	AI 2 I+ /RTD	AI 4 I+ /RTD	AI 6 I+ /RTD
7	AI 0 I- /RTD	AI 2 I- /RTD	AI 4 I- /RTD	AI 6 I- /RTD
8	AI 1 M+ /RTD	AI 3 M+ /RTD	AI 5 M+ /RTD	A7 M+ /RTD
9	AI 1 M- /RTD	AI 3 M- /RTD	AI 5 M- /RTD	AI 7 M- /RTD
10	AI 1 I+ /RTD	AI 3 I+ /RTD	AI 5 I+ /RTD	AI 7 I+ /RTD
11	AI 1 I- /RTD	AI 3 I- /RTD	AI 5 I- /RTD	AI 7 I- /RTD

Примечание

Неиспользуемые каналы для RTD могут быть отключены. Отключение неиспользуемых каналов не приводит к ошибкам.

Модуль RTD должен иметь непрерывную токовую петлю, чтобы устранить дополнительное время стабилизации, которое автоматически добавляется к неиспользованному каналу, который не деактивирован. Для стабильности у модуля RTD должно быть подключенное сопротивление (напр., 2-проводное подключение RTD).

A.11.2.1 Таблицы выбора для SM 1231 RTD

Таблица A-182 Диапазоны и точность для различных датчиков, поддерживаемых модулем RTD

Температурный коэффициент	Тип RTD	Минимальная нижняя граница температуры[1]	Номинальная нижняя граница температуры	Номинальная верхняя граница температуры	Максимальная верхняя граница температуры[2]	Погрешность преобразования при 25 °C	Погрешность преобразования при -20 °C до 60 °C
Pt 0,003850 ITS90 DIN EN 60751	Pt 100 climatic	-145,00 °C	-120,00 °C	145,00 °C	155,00 °C	±0,20 °C	±0,40 °C
	Pt 10	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±1,0 °C	±2,0 °C
	Pt 50	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±0,5 °C	±1,0 °C
	Pt 100						
	Pt 200						
	Pt 500						
	Pt 1000						
Pt 0,003902 Pt 0,003916 Pt 0,003920	Pt 100	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±0,5 °C	±1,0 °C
	Pt 200	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±0,5 °C	±1,0 °C
	Pt 500						
	Pt 1000						

Температурный коэффициент	Тип RTD	Минимальная нижняя граница температуры[1]	Номинальная нижняя граница температуры	Номинальная верхняя граница температуры	Максимальная верхняя граница температуры[2]	Погрешность преобразования при 25 °C	Погрешность преобразования при -20 °C до 60 °C
Pt 0,003910	Pt 10	-273,2 °C	-240,0 °C	1100,0 °C	1295 °C	±1,0 °C	±2,0 °C
	Pt 50	-273,2 °C	-240,0 °C	1100,0 °C	1295 °C	±0,8 °C	±1,6 °C
	Pt 100						
	Pt 500						
Ni 0,006720 Ni 0,006180	Ni 100	-105,0 °C	-60,0 °C	250,0 °C	295,0 °C	±0,5 °C	±1,0 °C
	Ni 120						
	Ni 200						
	Ni 500						
	Ni 1000						
LG-Ni 0,005000	LG-Ni 1000	-105,0 °C	-60,0 °C	250,0 °C	295,0 °C	±0,5 °C	±1,0 °C
Ni 0,006170	Ni 100	-105,0 °C	-60,0 °C	180,0 °C	212,4 °C	±0,5 °C	±1,0 °C
Cu 0,004270	Cu 10	-240,0 °C	-200,0 °C	260,0 °C	312,0 °C	±1,0 °C	±2,0 °C
Cu 0,004260	Cu 10	-60,0 °C	-50,0 °C	200,0 °C	240,0 °C	±1,0 °C	±2,0 °C
	Cu 50	-60,0 °C	-50,0 °C	200,0 °C	240,0 °C	±0,6 °C	±1,2 °C
	Cu 100						
Cu 0,004280	Cu 10	-240,0 °C	-200,0 °C	200,0 °C	240,0 °C	±1,0 °C	±2,0 °C
	Cu 50	-240,0 °C	-200,0 °C	200,0 °C	240,0 °C	±0,7 °C	±1,4 °C
	Cu 100						

[1] Значения от RTD ниже минимальной нижней границы температуры возвращаются как -32768.

[2] Значения от RTD выше максимальной верхней границы температуры возвращаются как +32767.

Таблица A-183 Сопротивление

Диапазон	Минимальная нижняя граница сопротивления	Номинальная нижняя граница сопротивления	Номинальная верхняя граница сопротивления	Максимальная верхняя граница сопротивления[1]	Погрешность преобразования при 25 °C	Погрешность преобразования при -20 °C до 60 °C
150 Ом	Неприменимо	0 (0 Ом)	27648 (150 Ом)	176.383 Ом	±0.05%	±0.1%
300 Ом	Неприменимо	0 (0 Ом)	27648 (300 Ом)	352.767 Ом	±0.05%	±0.1%
600 Ом	Неприменимо	0 (0 Ом)	27648 (600 Ом)	705.534 Ом	±0.05%	±0.1%

[1] Значения сопротивления выше максимальной верхней границы сопротивления возвращаются как +32767.

Примечание

Модуль возвращает 32767 для любого активированного канала без подключенного датчика. Если также разрешено обнаружение обрыва провода, модуль активирует соответствующие красные светодиоды.

Если используются RTD диапазоны 500 и 1000 Ом с другими сопротивлениями меньшей величины, ошибка может увеличиться вдвое от указанного значения ошибки.

Для 10 Ом диапазонов тепловых сопротивлений оптимальная точность будет достигнута при использовании 4-проводного подключения.

Сопротивление соединительных проводов при 2-проводном подключении вызовет ошибку интерпретации показаний датчика. В этом случае точность измерения не гарантируется.

Таблица A-184 Подавление помех и время обновления для модулей RTD

Выбор частоты подавления помех	Время интегрирования	Время обновления (сек)	
		4-канальный модуль	8-канальный модуль
400 Гц (2,5 мс)	10 мс [1]	4-/2-проводная схема: 0.142 3-проводная схема: 0.285	4-/2-проводная схема: 0.285 3-проводная схема: 0.525
60 Гц (16,6 мс)	16,67 мс	4-/2-проводная схема: 0.222 3-проводная схема: 0.445	4-/2-проводная схема: 0.445 3-проводная схема: 0.845
50 Гц (20 мс)	20 мс	4-/2-проводная схема: 0.262 3-проводная схема: .505	4-/2-проводная схема: 0.524 3-проводная схема: 1.015
10 Гц (100 мс)	100 мс	4-/2-проводная схема: 1.222 3-проводная схема: 2.445	4-/2-проводная схема: 2.425 3-проводная схема: 4.845

[1] Для того, чтобы достичь заявленной точности и разрешения для модуля при фильтре 400 Гц, время интегрирования составляет 10 мс. Этот фильтр также отсекает помехи с частотой 100 Гц и 200 Гц.

Примечание

После включения модуль выполняет внутреннюю калибровку аналого-цифрового преобразователя. В течение этого времени модуль возвращает значение 32767 в каждом канале, пока действительные данные не станут доступны на этом канале. При необходимости следует учесть это время инициализации в программе пользователя. Поскольку конфигурация модуля может изменить продолжительность времени инициализации, следует проверить поведение модуля в конкретной конфигурации. При необходимости можно включать логику в свою программу пользователя, чтобы подстроиться под время инициализации модуля.

Представление аналоговых значений для RTD

В таблице ниже представлено оцифрованное измеренное значение для датчиков со стандартным диапазоном температур RTD.

Таблица A-185 Представление аналоговых значений для термометров сопротивлений PT 100, 200, 500, 1000 и PT 10, 50, 100, 500 GOST (0,003850) стандарт

Pt x00 стандарт в °C (1 цифра = 0,1 °C)	**Единицы**		**Pt x00 стандарт в °F (1 цифра = 0,1 °F)**	**Единицы**		**Диапазон**
	Десятичные	**Шестнадцатеричные**		**Десятичные**	**Шестнадцатеричные**	
> 1000.0	32767	7FFF	> 1832.0	32767	7FFF	Выход за верхний установленный предел
1000.0 : 850.1	10000 : 8501	2710 : 2135	1832.0 : 1562.1	18320 : 15621	4790 : 3D05	Выше диапазона
850.0 : -200.0	8500 : -2000	2134 : F830	1562.0 : -328.0	15620 : -3280	3D04 : F330	Расчетный диапазон
-200.1 : -243.0	-2001 : -2430	F82F : F682	-328.1 : -405.4	-3281 : -4054	F32F : F02A	Ниже диапазона
< -243.0	-32768	8000	< -405.4	-32768	8000	Выход сигнала за нижний установленный предел

A.12 Технологические модули

A.12.1 SM 1278 4xIO-Link-Master SM

Таблица A-186 Общие технические данные

Технические данные		**Сигнальный модуль SM 1278 4xIO-Link-Master**
Заказной номер		6ES7278-4BD32-0XB0
Габаритные размеры (Ш x В x Г) (мм)		45 x 100 x 75
Вес		150 г
Общая информация		
	Данные идентификации и обслуживания (I&M)	Да, IM0 до IM3
Напряжение питания		

Технические данные			**Сигнальный модуль SM 1278 4xIO-Link-Master**
	Номинальное напряжение (DC)		24 В DC
	Нижнее предельное значение допустимого диапазона (DC)		19,2 В; 20,5 В если используется IO-Link (напряжение питания для устройств IO-Link на ведущем устройстве не должно быть ниже 20 В)
	Верхнее предельное значение допустимого диапазона (DC)		28,8 В DC
	Защита от неправильной полярности		Да
Входной ток			
	Потребляемый ток		65 мА, без нагрузки
Электропитание энкодера			
	Количество выходов		4
	Выходной ток, ном. значение		200 мА
Мощность потерь			
	Мощность потерь, тип.		1 Вт, без нагрузки порта
Цифровые входы/выходы			
	Длина кабеля (в метрах)		Макс. 20 м, неэкранированный
SDLC			
	Длина кабеля (в метрах)		Макс. 20 м, неэкранированный
IO-Link			
	Число портов		4
	Число портов, одновременно находящихся в активном состоянии		4
	IO-Link протокол 1.0		Да
	IO-Link протокол 1.1		Да
Рабочее состояние			
	IO-Link		Да
	DI		Да
	DO		Да, макс. 100 мА
Подключение IO-Link устройств			
	Порт A		Да
		Скорость передачи данных	4,8 кбод (COM1)
			38,4 кбод (COM2)
			230,4 кбод (COM3)
	Мин. время цикла		2 мс, зависит от объема данных пользователя
	Объем данных, ввод на порт		макс. 32 байта
	Объем данных, ввод на модуль		32 байта
	Объем данных, вывод на порт		макс. 32 байта
	Объем данных, вывод на модуль		32 байта
	Объем памяти для параметров настройки		2 кбайта
	Max. длина неэкранированного кабеля (в метрах)		20 м
Состояния, прерывания, диагностика			
	Индикация состояния		Да
Прерывания			
	Диагностическое прерывание		Да; диагностика порта доступна только в режиме IO-Link

Технические данные			**Сигнальный модуль SM 1278 4xIO-Link-Master**
Диагностические прерывания			
	Диагностика		
		Контроль напряжения питания	Да
		Короткое замыкание	Да
Диагностический светодиод индикации			
	Контроль напряжения питания		Да, мигающий красный светодиод DIAG
	Индикация состояния канала		Да, один зеленый светодиод Qn на канал в стандартном режиме работы и один зеленый светодиод Cn на канал в режиме IO-Link
	Для диагностики канала		Да, красный светодиод Fn
	Для диагностики модуля		Да, зеленый/ красный светодиод DIAG
Гальваническая развязка			
	Гальваническое разделение каналов		
		Между каналами	Нет
		Между каналами и внутренней шиной	Да
Изоляция			
	Испытательное напряжение для изоляции		707 В DC (типовое испытание)
Условия окружающей среды			
	Рабочая температура		
		Мин.	-20 °C
		Макс.	60 °C
		Горизонтальная установка, мин.	-20 °C
		Горизонтальная установка, макс.	60 °C
		Вертикальная установка, мин.	-20 °C
		Вертикальная установка, макс.	50 °C

Обзор времени отклика

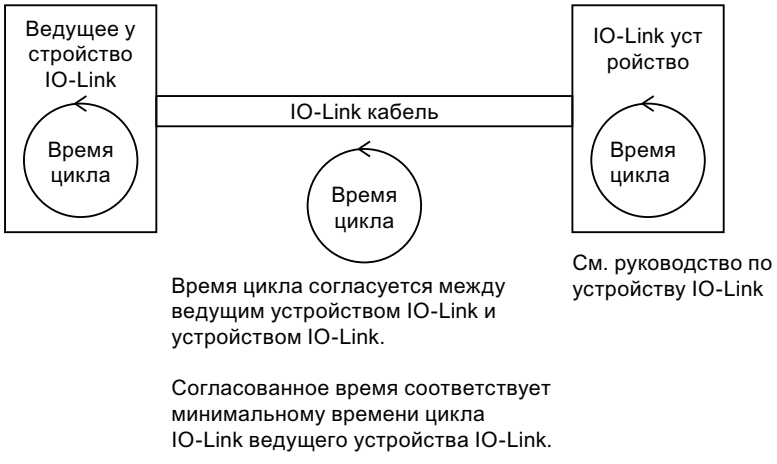


Таблица A-187 Схема электрических соединений для SM 1278 IO-Link-Master

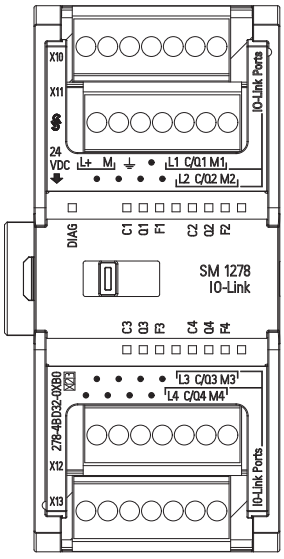
SM 1278 IO-Link-Master (6ES7278-4BD32-0XB0)	
	

Таблица A-188 Разводка контактов для SM 1278 IO-Link-Master (6ES7278-4BD32-0XB0)

Контакт	X10	X11	X12	X13
1	L+ / 24 В DC	Не подключено	Не подключено	Не подключено
2	M / 24 В DC	Не подключено	Не подключено	Не подключено
3	Функциональное заземление	Не подключено	Не подключено	Не подключено
4	Не подключено	Не подключено	Не подключено	Не подключено
5	L_1	L_2	L_3	L_4
6	C/O_1	C/O_2	C/O_3	C/O_4
7	M_1	M_2	M_3	M_4

A.12.1.1 Обзор модуля SM 1278 4xIO-Link-Master

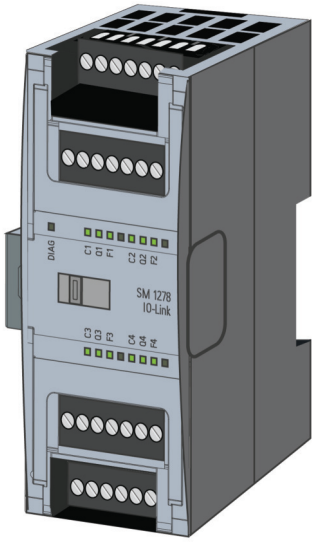
SM 1278 4xIO-Link-Master - это модуль с 4 портами, который работает и как сигнальный модуль, и как коммуникационный модуль. Каждый порт может работать в IO-Link режиме, как одиночный цифровой вход 24 В DC или как цифровой выход на 24 В DC.

Ведущее устройство IO-Link программирует ациклический обмен данными с устройством IO-Link с помощью функционального блока (FB) IO_LINK_DEVICE в программе STEP 7 контроллера S7-1200. FB IO_LINK_DEVICE указывает, какое ведущее устройство IO-Link использует в программе, и какие порты ведущее устройство использует для обмена данными

На веб-сайте Онлайн-поддержки для промышленности Siemens (http://support.industry.siemens.com) можно найти подробную информацию о работе с

библиотекой IO-Link . Ввести "IO-Link" в поисковой строке веб-сайта, чтобы получить доступ к информации о продуктах IO-Link и их использовании.

Вид модуля



Свойства

Технические характеристики

- Ведущее устройство IO-Link согласно спецификации IO-Link V1.1 (см веб-сайт объединения IO-Link (http://io-link.com/en/index.php) для дополнительной информации)
- Последовательный коммуникационный модуль с четырьмя портами (каналами)
- Скорость передачи данных COM1 (4.8 kbaud), COM2 (38.4 kbaud), COM3 (230.4 kbaud)
- SIO режим (стандартный режим ввода-вывода)
- Подключение до четырех устройств IO-Link (3-проводное соединение) или четырех стандартных исполнительных устройств или стандартных энкодеров
- Программируемая функция диагностики для каждого порта

Поддерживаемые функции

- Идентификационные данные для I&M (идентификация и обслуживание)
- Обновление прошивки
- Параметрирование IO-Link с помощью инструмента конфигурации портов S7-PCT, STEP 7 Professional и S7-1200 V4.0 или выше CPU. . STEP 7 Professional V15 поддерживает параметрирование IO-Link (с помощью V2.1 HSP или выше) в TIA Portal с ограниченным объемом функций.

- Биты PQI (Port Qualifier Information)
- Резервное копирование и восстановление FB библиотеки IO-Link

IO-Link это соединение точка-точка между ведущим и ведомым устройством. Как обычные, так и интеллектуальные датчики/исполнительные механизмы, могут использоваться в качестве устройств на IO-Link через неэкранированные стандартные кабели, используя проверенную 3-проводную технологию. IO-Link обратно совместима с обычными цифровыми датчиками и исполнительными элементами. Структура схемы и канал данных разработаны согласно испытанной технологии 24 V DC.

Для получения дополнительной информации относительно технологии SIMATIC-IO-Link можно обратиться к "Справочнику по системе "IO-Link" на веб-сайте Онлайн-поддержки для промышленности Siemens (http://support.automation.siemens.com)

Примечание

Данные параметров IO-Link

При замене модуля SM 4xIO-Link Master, данные параметров не назначаются ему автоматически.

⚠ ОСТОРОЖНО

Извлечение и вставка

Если модуль SM 4xIO-Link Master вставляется при включенной нагрузке, это может привести к возникновению опасных состояний на установке.

Возможно повреждение системы автоматизации S7-1200.

Извлечение и вставка модуля SM 4xIO-Link Master должно выполняться только при отключенной нагрузке.

Эффект сброса на заводские установки

Функция "Сброс на заводские установки" используется для восстановления параметров, назначенных с помощью S7-PCT, к состоянию при поставке.

После" Сброса на заводские установки " параметры модуля SM 1278 4xIO-Linkбудут установлены следующим образом:

- Порты находятся в режиме DI
- Порты привязаны к относительным адресам от 0.0 до 0.3
- PortQualifier отменен
- Данные обслуживания с 1 до 3 удалены

Примечание

При сбросе на заводские установки, параметры устройства удаляются, и состояние при поставке восстанавливается.

Перед удалением сигнального модуля SM 1278 4xIO-Link, следует выполнить сброс на заводские установки прежде, чем поместить его на хранение.

Порядок действий

Для "сброса на заводские установки" использовать описание в интерактивной справке для S7-PCT на вкладке Конфигурация ведущего устройства > вкладка 'Команды'.

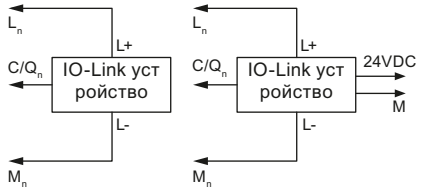
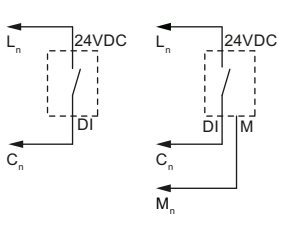
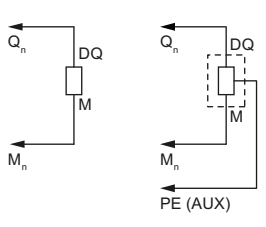
A.12.1.2 Подключение

Подробную информацию о разводке контактов можно найти в таблице разводки контактов для SM 1278 I/O-Link-Master (6ES 278-4BD32-0XB0). (Страница 1563)

В следующей таблице показана разводка контактов для SM 1278 4xIO-Link-Master:

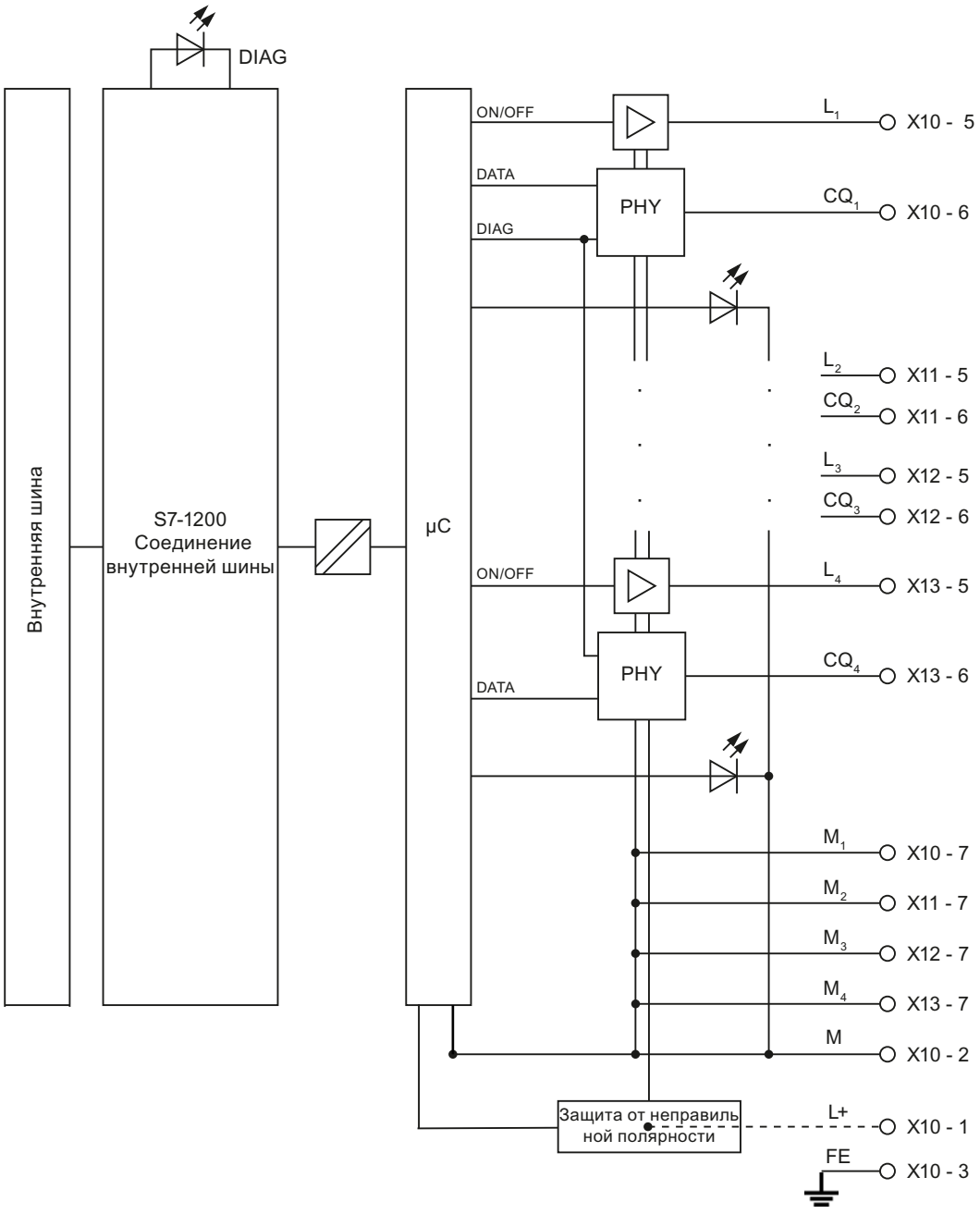
Контакт	X10	X11	X12	X13	Примечания	Базовый блок
7	M_1	M_2	M_3	M_4	• M_n: Масса к ведомому устройству • C/O_n: SDLC, DI или DO • L_n: 24 В DC к ведомому устройству • M: земля • L+: 24 В DC к ведущему устройству • RES: зарезервировано; не занимать	A1
6	C/O_1	C/O_2	C/O_3	C/O_4		
5	L_1	L_2	L_3	L_4		
4	RES	RES	RES	RES		
3	⏚ (функциональное заземление)	RES	RES	RES		
2	M	RES	RES	RES		
1	L+	RES	RES	RES		

В приведенной ниже таблице показаны примеры подключения, n = номер порта.

Режим IO-Link	Режим DI	Режим DO
3-проводное 5-проводное 	2-жильное 3-жильное 	2-жильное 3-жильное 

Примечание

Подключенные датчики должны использовать источник питания посредством соединения L_n с ведущим модулем.



A.12.1.3 Параметры/адресное пространство

Конфигурирование модуля SM 1278 4xIO-Link Master

Для интеграции модуля параметрирования и ввода в эксплантацию потребуется STEP 7 V13 или выше. Для выполнения определенных функций также потребуется S7-PCT (Port Configuration Tool).

В приведенной ниже таблице показан, когда нужен инструмент S7-PCT:

	SM 1278 V2.0 4xIO-Link Master	**SM 1278 V2.1 4xIO-Link Master**
STEP7 V13.x и STEP V14.x	Нужен S7-PCT	Нужен S7-PCT
STEP 7 15.x	Нужен S7-PCT	S7-PCT не нужен для базовых функций S7-PCT не нужен для расширенных функций

Дополнительная информация находится в справочнике по системе SIMATIC IO-Link (https://support.industry.siemens.com/cs/ww/en/view/65949252).

В следующей таблице показаны параметры для SM 1278 4xIO-Link-Master:

Параметры	**Диапазон значений**	**По умолчанию**	**Конфигурирование в RUN**	**Область действия**
Диагностический порт 1	• Выключено • Включено	Выключено	Да	Порт (канал)
Диагностический порт 2	• Выключено • Включено	Выключено	Да	Порт (канал)
Диагностический порт 3	• Выключено • Включено	Выключено	Да	Порт (канал)
Диагностический порт 4	• Выключено • Включено	Выключено	Да	Порт (канал)

Параметр для разрешения диагностики портов с 1 по 4

Этот параметр позволяет включить диагностику для конкретных портов из имеющихся четырех портов IO-Link.

Назначения портов следующие:

Порт 1 → Канал 1

Порт 2 → Канал 2

Порт 3 → Канал 3

Порт 4 → Канал 4

Максимальный размер адресов ввода и вывода модуля SM 4xIO-Link Master составляет 32 байта в каждом случае. Для назначения диапазонов адресов можно использовать инструмент конфигурации для портов S7-PCT или, начиная с версии V15 V2.1 HSP или выше, конфигурацию оборудования в TIA Portal.

Набор параметров

Параметрирование в программе пользователя

Можно конфигурировать устройство во времени работы.

Изменение параметров во время работы

Параметры модуля включены в набор данных 128. Можно передать редактируемые параметры в модуль с помощью инструкции WRREC.

После сброса (выключить и включить) CPU, CPU заменяет параметры, которые были переданы при параметрировании с помощью инструкции WRREC в модуль.

Инструкция для параметрирования

Для параметрирования модуля ввода-вывода в программе пользователя предлагается следующая инструкция:

Инструкция	Применение
SFB 53 WRREC	Передача изменяемых параметров в модуль.

Сообщение об ошибке

Следующее возвращаемое значение передается в сообщении об ошибке:

Код ошибки	Значение
$80B1_H$	Ошибка длины данных
$80E0_H$	Ошибка в заголовке информации
$80E1_H$	Ошибка параметра

Структура набора данных

В приведенной ниже таблице показаны параметры IO-Link:

Смещение	Обозначение	Тип	По умолчанию	Описание
0	Версия	1 байт	0x02	Показывает структуру набора данных 0x02 IO-Link-Master поIO-Link V1.1
1	Длина параметра	1 байт	0x02	Длина параметра (2 байта + 2 заголовка)
Стартовые параметры IO-Link				
2	Диагностика порта (порт1 1 до n)	1 байт	0x00	Включение диагностики для порта 1 до n
3	Свойства IOL	1 байт	0x00	Свойства модуля

В следующей таблице показана версия набора данных:

Бит 7	**Бит 6**	**Бит 5**	**Бит 4**	**Бит 3**	**Бит 2**	**Бит 1**	**Бит 0**
Зарезервировано		Основная версия (00)		Дополнительная версия (0010)			

В следующей таблице показан набор данных для диагностики порта:

Бит 7	**Бит 6**	**Бит 5**	**Бит 4**	**Бит 3**	**Бит 2**	**Бит 1**	**Бит 0**
Зарезервировано				EN_Port4	EN_Port3	EN_Port2	EN_Port1

EN_Portx:

0 = диагностика деактивирована

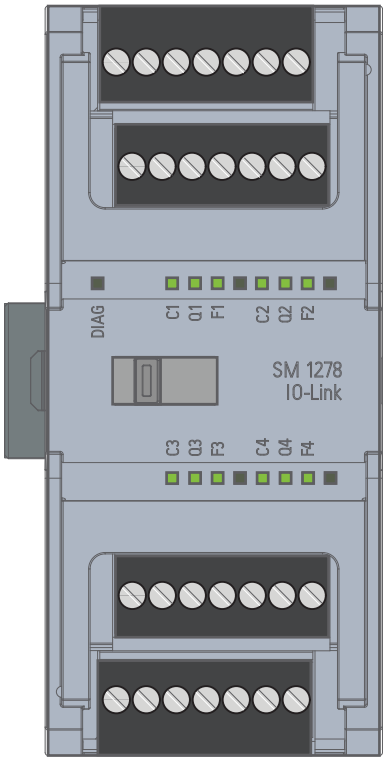
1 = диагностика активирована

В следующей таблице показан набор данных со свойствами IOL:

Бит 7	**Бит 6**	**Бит 5**	**Бит 4**	**Бит 3**	**Бит 2**	**Бит 1**	**Бит 0**
Зарезервировано							

A.12.1.4 Прерывания, сообщения об ошибках и системные события

Светодиодный индикатор



Значение светодиодных индикаторов

Следующая таблица поясняет значение индикаторов ошибок и состояния. Методы по устранению для диагностических тревог можно найти в разделе "Диагностические тревоги".

Светодиод DIAG

DIAG	Значение
☐ Выкл	Питание внутренней шины S7-1200 не в порядке
☼ Мигает	Модуль не сконфигурирован

DIAG	Значение
■ Вкл	Параметры модуля заданы и отсутствуют условия для диагностики модуля
 Мигает	Параметры модуля заданы и выполняется диагностика модуля ИЛИ L+ не подключено

Светодиод состояния порта

Для IO-Link порта в режиме IO-Link.

COM/1 ... COM/4	Значение
□ Выкл	Порт деактивирован
 Мигает	PORT активирован, устройство не подключено или Порт не подключен к сконфигурированному устройству
■ Вкл	PORT активирован, устройство подключено

Светодиод состояния канала

Для IO-Link порта в DI/Q-модуле.

DI/Q1 ... DI/Q4	Значение
□ Выкл	Технологический сигнал = 0
■ Вкл	Технологический сигнал = 1

Светодиод ошибки порта

F1...F4	Значение
□ Выкл	Ошибки отсутствуют
■ Вкл	Ошибка

Ошибки модуля индицируются как диагностика (состояние модуля) только IO-Link режиме.

Диагностическое прерывание	**Код ошибки (десятичный формат)**	**STATUS (W#16#...)**	**Значение (код ошибки IO-Link)**	**Ведущее устройство IO-Link**	**Устройство IO-Link**
Короткое замыкание	1	1804	Короткое замыкание в технологических кабелях устройства IO-Link	X	
		7710	Короткое замыкание в устройстве ввода-вывода		X
Напряжение ниже допустимого	2	5111 5112	Слишком низкое напряжение питания		X
Напряжение выше допустимого	3	5110	Слишком высокое напряжение питания		X
Перегрев	5	1805	Температура на ведущем устройстве превышена	X	
		4000 4210	Температура на устройстве превышена		X
Обрыв провода	6	1800	• Устройство IO-Link не подключено • Обрыв сигнального кабеля к устройству IO-Link • Обмен данными с устройством IO-Link невозможен из-за другой ошибки	X	
Выход за верхний установленный предел	7	8C10 8C20	Выход за пределы максимального значения технологической переменной (тега)		X
		8C20	Превышение диапазона измерения		
Выход сигнала за нижний установленный предел	8	8C30	Выход за пределы минимального значения технологической переменной (тега)		X
Ошибка	9	---	Все отсутствующие в этом списке коды ошибок IO-Link относятся к этой ошибке PROFIBUS DP.		X
Ошибка параметрирования	16	1882 1883	Конфигурирование ведущего устройства IO-Link невозможно	X	
		1802	Неправильное устройство		
		1886	Ошибка сохранения		
		6320 6321 6350	Устройство было сконфигурировано неправильно		X

Диагностическое прерывание	Код ошибки (десятичный формат)	STATUS (W#16#...)	Значение (код ошибки IO-Link)	Ведущее устройство IO-Link	Устройство IO-Link
Напряжение питания отсутствует	17	1806	Отсутствует напряжение питания для устройства L+	X	
		1807	Напряжение питания для устройства L+ слишком мало (<20 В)		
Неисправный предохранитель	18	5101	Предохранитель в устройстве неисправен		X
Аварийное отключение	25	1880	Серьезная ошибка (требуется замена ведущего устройства)	X	
Внешняя ошибка	26	1809 180A 180B 180C 180D	Ошибка в памяти данных	X	
		1808	Больше 6 активных ошибок одновременно на устройстве IO-Link		

A.13 Цифровые сигнальные платы (SB)

A.13.1 Технические данные сигнальной платы цифрового ввода SB 1221 200 кГц

Таблица A-189 Общие технические данные

Технические данные	SB 1221 DI 4 x 24 VDC, 200 кГц	SB 1221 DI 4 x 5 VDC, 200 кГц
Заказной номер	6ES7221-3BD30-0XB0	6ES7221-3AD30-0XB0
Габаритные размеры (Ш x В x Г) (мм)	38 x 62 x 21	
Вес	35 г	
Рассеиваемая мощность	1,5 Вт	1,0 Вт
Потребляемый ток (SM шина)	40 мА	
Потребляемый ток (24 В DC)	7 мА / вход + 20 мА	15 мА / вход + 15 мА

Таблица A-190 Цифровые входы

Технические данные	SB 1221 DI 4 x 24 VDC, 200 кГц	SB 1221 DI 4 x 5 VDC, 200 кГц
Количество входов	4	
Тип	Источник	
Номинальное напряжение	24 В DC при 7 мА, ном.	5 В DC при 15 мА, ном.

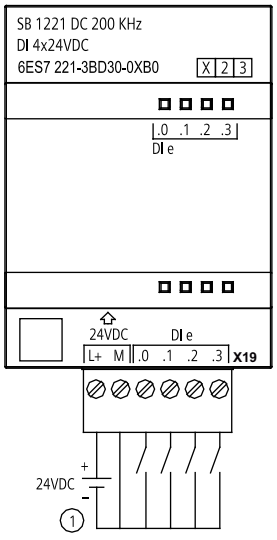
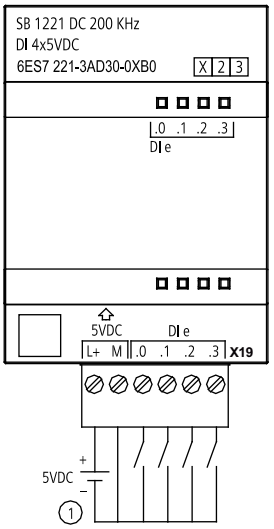
Технические данные		**SB 1221 DI 4 x 24 VDC, 200 кГц**	**SB 1221 DI 4 x 5 VDC, 200 кГц**
Длительно допустимое напряжение		28,8 В DC	6 В DC
Перенапряжение		35 В DC на 0,5 с	6 В
Сигнал логический 1		0 В (10 mA) на L+ минус 10 В (2,9 мА)	0 В (20 mA) на L+ минус 2,0 В (5,1 мА)
Сигнал логический 0		L+ минус 5 В (1,4 мА) на L+ (0 мА)	L+ минус 1,0 В (2,2 мА) на L + (0 мА)
Скорость изменения входного сигнала высокоскоростных счетчиков (макс.)		Частота следования 1-фазных входных сигналов: 200 кГц Частота следования 2-фазных входных сигналов: 160 кГц	
Электрическое разделение (между полевой стороной и логикой)		707 В DC (типовое испытание)	
Изолированные группы		1	
Время фильтрации	мкс установка:	0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0	
	мс установка:	0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0	
Количество одновременно активных входов		• 2 (без смежных точек) при 60 °C для горизонтальной установки или 50 °C для вертикальной установки • 4 при 55 °C для горизонтальной установки или 45 °C для вертикальной установки	4
Длина кабеля (в метрах)		50 м, экранированная витая пара	

Примечание

При значении частоты переключения выше 20 кГц, важно, чтобы цифровые входы принимали прямоугольный сигнал. Рассмотрим следующие варианты улучшения качества сигнала на входах:

- Минимизировать длину кабеля.
- Сменить тип драйвера с потребляющего ток на потребляющего/отдающего ток.
- Выбрать кабель лучшего качества.
- Перейти от схемы/компонентов 24 В на 5 В.
- Подключить внешнюю нагрузку на входе.

Таблица A-191 Схемы электрических соединений для SB цифрового ввода (200 кГц)

SB 1221 DI 4 x 24 VDC, 200 кГц (6ES7221-3BD30-0XB0)	**SB 1221 DI 4 x 5 VDC, 200 кГц (6ES7221-3AD30-0XB0)**
	

① Поддерживает только входы-источники

Таблица A-192 Разводка контактов для SB 1221 DI 4 x 24 VDC, 200 кГц (6ES7221-3BD30-0XB0)

Контакт	**X19**
1	L+ / 24 В DC
2	M / 24 В DC
3	DI e.0
4	DI e.1
5	DI e.2
6	DI e.3

Таблица A-193 Разводка контактов для SB 1221 DI 4 x 5 VDC, 200 кГц (6ES7221-3AD30-0XB0)

Контакт	**X19**
1	L+ / 5 В DC
2	M / 5 В DC
3	DI e.0
4	DI e.1
5	DI e.2
6	DI e.3

A.13.2 Технические данные сигнальной платы цифрового вывода SB 1222 200 кГц

Таблица A-194 Общие технические данные

Технические данные	SB 1222 DQ 4 x 24 VDC, 200 кГц	SB 1222 DQ 4 x 5 VDC, 200 кГц
Заказной номер	6ES7222-1BD30-0XB0	6ES7222-1AD30-0XB0
Габаритные размеры (Ш x В x Г) (мм)	38 x 62 x 21	
Вес	35 г	
Рассеиваемая мощность	0,5 Вт	
Потребляемый ток (SM шина)	35 мА	
Потребляемый ток (24 В DC)	15 мА	

Таблица A-195 Цифровые выходы

Технические данные	SB 1222 DQ 4 x 24 VDC, 200 кГц	SB 1222 DQ 4 x 5 VDC, 200 кГц
Выходы	4	
Тип выхода	Двунаправленные транзисторные ключи MOSFET[1]	
Диапазон напряжений	от 20,4 до 28,8 В DC	от 4,25 до 6,0 В DC
Сигнал логический 1 при макс. токе	L+ - 1,5 В	L+ - 0,7 В
Сигнал логический 0 при макс. токе	Макс. 1,0 В DC	Макс. 0,2 В DC
Ток (макс.)	0,1 А	
Ламповая нагрузка	--	
Сопротивление активного выходного канала	Макс. 11 Ом	Макс. 7 Ом
Сопротивление в неактивном состоянии	Макс. 6 Ом	Макс. 0,2 Ом
Ток утечки на выход	--	
Частота следования импульсов	макс. 200 кГц, мин. 2 Гц	
Импульсный ток нагрузки, не более	0,11 А	
Защита от перегрузки	Нет	
Электрическое разделение (между полевой стороной и логикой)	707 В DC (типовое испытание)	
Изолированные группы	1	
Ток на общий провод	0,4 А	
Индуктивное фиксирующее напряжение	Нет	
Задержка переключения	1,5 мкс + 300 нс от низкого к высокому уровню 1,5 мкс+ 300 нс от высокого к низкому уровню	200 нс + 300 нс от низкого к высокому уровню 200 нс+ 300 нс от высокого к низкому уровню
Поведение при переходе из RUN в STOP	Последнее значение или замещающее значение (по умолчанию 0)	

Технические данные	SB 1222 DQ 4 x 24 VDC, 200 кГц	SB 1222 DQ 4 x 5 VDC, 200 кГц
Количество одновременно активных выходов	• 2 (без смежных точек) при 60 °C для горизонтальной установки или 50 °C для вертикальной установки • 4 при 55 °C для горизонтальной установки или 45 °C для вертикальной установки	4
Длина кабеля (в метрах)	50 м, экранированная витая пара	

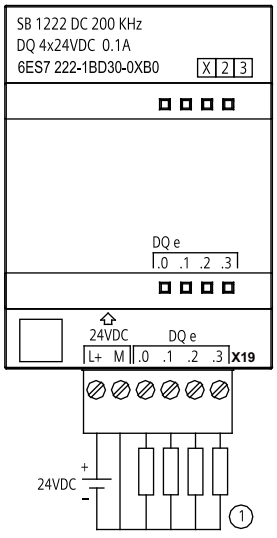
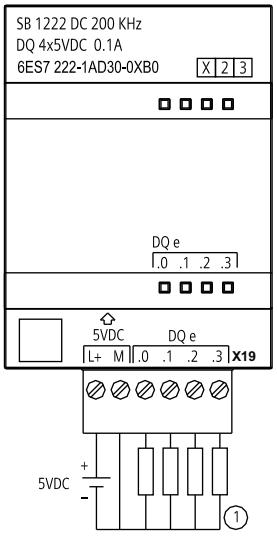
[1] Поскольку как потребляющие, так и отдающие ток конфигурации поддерживаются той же самой схемой, активное состояние отдающей ток нагрузки противоположно активному состоянию потребляющей нагрузки. Отдающий ток выход проявляет положительную логику (Q бит и светодиод активны, когда течет ток в нагрузке), в то время как потребляющий ток выход проявляет отрицательную логику (Q бит и светодиод неактивны, когда есть ток в нагрузке). Если модуль вставлен без пользовательской программы, на выходе модуля по умолчанию будет 0 В, то есть потребляющая ток нагрузка будет включена.

Примечание

При значении частоты переключения выше 20 кГц, важно, чтобы цифровые входы принимали прямоугольный сигнал. Рассмотрим следующие варианты улучшения качества сигнала на входах:

- Минимизировать длину кабеля.
- Сменить тип драйвера с потребляющего ток на потребляющего/отдающего ток.
- Выбрать кабель лучшего качества.
- Перейти от схемы/компонентов 24 В на 5 В.
- Подключить внешнюю нагрузку на входе.

Таблица A-196 Схемы электрических соединений для SB цифрового вывода (200 кГц)

SB 1222 DQ 4 x 24 VDC, 200 кГц (6ES7222-1BD30-0XB0)	**SB 1222 DQ 4 x 5 VDC, 200 кГц (6ES7222-1AD30-0XB0)**
	

① Для отдающих ток выходов подключить "Нагрузку" к "-" (показано на рис.). Для потребляющих ток выходов подключить "Нагрузку" к "+". Поскольку как потребляющие, так и отдающие ток конфигурации поддерживаются той же самой схемой, активное состояние отдающей ток нагрузки противоположно активному состоянию потребляющей нагрузки. Отдающий ток выход проявляет положительную логику (Q бит и светодиод активны, когда течет ток в нагрузке), в то время как потребляющий ток выход проявляет отрицательную логику (Q бит и светодиод неактивны, когда есть ток в нагрузке). Если модуль вставлен без пользовательской программы, на выходе модуля по умолчанию будет 0 В, то есть потребляющая ток нагрузка будет включена.

Примечание

Проследить, чтобы провод соединения с корпусом был правильно заземлен. Потеря соединения с землей для высокоскоростных сигнальных плат цифрового вывода может вызвать ток утечки в объеме, достаточном для активации нагрузки постоянного тока. Если выходы используются для важных нагрузок постоянного тока, следует соблюдать особую осторожность, используя резервный заземляющий провод к SB.

Таблица A-197 Разводка контактов для SB 1222 DQ 4 x 24 VDC, 200 кГц (6ES7222-1BD30-0XB0)

Контакт	X19
1	L+ / 24 В DC
2	M / 24 В DC
3	DQ e.0
4	DQ e.1
5	DQ e.2
6	DQ e.3

Таблица A-198 Разводка контактов для SB 1222 DQ 4 x 5 VDC, 200 кГц (6ES7222-1AD30-0XB0)

Контакт	X19
1	L+ / 5 В DC
2	M / 5 В DC
3	DQ e.0
4	DQ e.1
5	DQ e.2
6	DQ e.3

A.13.3 Технические данные сигнальной платы цифрового ввода/вывода SB 1223 200 кГц

Таблица A-199 Общие технические данные

Технические данные	SB 1223 DI 2 x 24 VDC / DQ 2 x 24 VDC, 200 кГц	SB 1223 DI 2 x 5 VDC / DQ 2 x 5 VDC, 200 кГц
Заказной номер	6ES7223-3BD30-0XB0	6ES7223-3AD30-0XB0
Габаритные размеры (Ш x В x Г) (мм)	38 x 62 x 21	
Вес	35 г	
Рассеиваемая мощность	1,0 Вт	0,5 Вт
Потребляемый ток (SM шина)	35 мА	
Потребляемый ток (24 В DC)	7 мА / вход + 30 мА	15 мА / вход + 15 мА

Таблица A-200 Цифровые входы

Технические данные	SB 1223 DI 2 x 24 VDC / DQ 2 x 24 VDC, 200 кГц	SB 1223 DI 2 x 5 VDC / DQ 2 x 5 VDC, 200 кГц
Количество входов	2	
Тип	Источник	
Номинальное напряжение	24 В DC при 7 мА, ном.	5 В DC при 15 мА, ном.
Длительно допустимое напряжение	28,8 В DC	6 В DC
Перенапряжение	35 В DC на 0,5 с	6 В
Сигнал логический 1	0 В (10 mA) на L+ минус 10 В (2,9 мА)	0 В (20 mA) на L+ минус 2,0 В (5,1 мА)
Сигнал логический 0	L+ минус 5 В (1,4 мА) на L+ (0 мА)	L+ минус 1,0 В (2,2 мА) на L + (0 мА)
Скорость изменения входного сигнала высокоскоростных счетчиков (макс.)	Частота следования 1-фазных входных сигналов: 200 кГц Частота следования 2-фазных входных сигналов: 160 кГц	
Электрическое разделение (между полевой стороной и логикой)	707 В DC (типовое испытание)	
Изолированные группы	1 (отсутствует развязка с выходом)	

Технические данные		**SB 1223 DI 2 x 24 VDC / DQ 2 x 24 VDC, 200 кГц**	**SB 1223 DI 2 x 5 VDC / DQ 2 x 5 VDC, 200 кГц**
Время фильтрации	мкс установка:	0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0	
	мс установка:	0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0	
Количество одновременно активных входов		2	
Длина кабеля (в метрах)		50 м, экранированная витая пара	

Таблица A-201 Цифровые выходы

Технические данные	**SB 1223 DI 2 x 24 VDC / DQ 2 x 24 VDC, 200 кГц**	**SB 1223 DI 2 x 5 VDC / DQ 2 x 5 VDC, 200 кГц**
Выходы	2	
Тип выхода	Двунаправленные транзисторные ключи MOSFET[1]	
Диапазон напряжений	от 20,4 до 28,8 В DC	от 4,25 до 6,0 В DC
Ном. значение	24 В DC	5 В DC
Сигнал логический 1 при макс. токе	L+ - 1,5 В	L+ - 0,7 В
Сигнал логический 0 при макс. токе	Макс. 1,0 В DC	Макс. 0,2 В DC
Ток (макс.)	0,1 А	
Ламповая нагрузка	--	
Сопротивление активного выходного канала	Макс. 11 Ом	Макс. 7 Ом
Сопротивление в неактивном состоянии	Макс. 6 Ом	Макс. 0,2 Ом
Ток утечки на выход	--	
Частота следования импульсов	макс. 200 кГц, мин. 2 Гц	
Импульсный ток нагрузки, не более	0,11 А	
Защита от перегрузки	Нет	
Электрическое разделение (между полевой стороной и логикой)	707 В DC (типовое испытание)	
Изолированные группы	1 (отсутствует развязка с входом)	
Ток на общий провод	0,2 А	
Индуктивное фиксирующее напряжение	Нет	
Задержка переключения	1,5 мкс + 300 нс от низкого к высокому уровню 1,5 мкс+ 300 нс от высокого к низкому уровню	200 нс + 300 нс от низкого к высокому уровню 200 нс+ 300 нс от высокого к низкому уровню
Поведение при переходе из RUN в STOP	Последнее значение или замещающее значение (по умолчанию 0)	

Технические данные	**SB 1223 DI 2 x 24 VDC / DQ 2 x 24 VDC, 200 кГц**	**SB 1223 DI 2 x 5 VDC / DQ 2 x 5 VDC, 200 кГц**
Количество одновременно активных выходов	2	
Длина кабеля (в метрах)	50 м, экранированная витая пара	

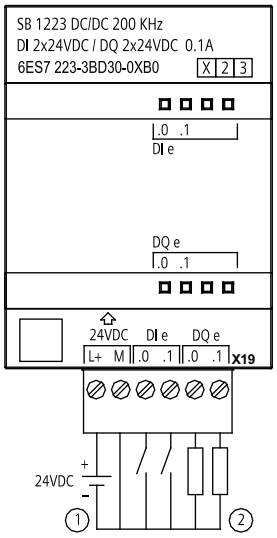
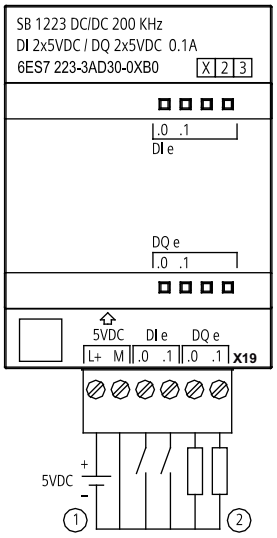
[1] Поскольку как потребляющие, так и отдающие ток конфигурации поддерживаются той же самой схемой, активное состояние отдающей ток нагрузки противоположно активному состоянию потребляющей нагрузки. Отдающий ток выход проявляет положительную логику (Q бит и светодиод активны, когда течет ток в нагрузке), в то время как потребляющий ток выход проявляет отрицательную логику (Q бит и светодиод неактивны, когда есть ток в нагрузке). Если модуль вставлен без пользовательской программы, на выходе модуля по умолчанию будет 0 В, то есть потребляющая ток нагрузка будет включена.

Примечание

При значении частоты переключения выше 20 кГц, важно, чтобы цифровые входы принимали прямоугольный сигнал. Рассмотрим следующие варианты улучшения качества сигнала на входах:

- Минимизировать длину кабеля.
- Сменить тип драйвера с потребляющего ток на потребляющего/отдающего ток.
- Выбрать кабель лучшего качества.
- Перейти от схемы/компонентов 24 В на 5 В.
- Подключить внешнюю нагрузку на входе.

Таблица A-202 Схемы электрических соединений для SB цифрового ввода/вывода (200 кГц)

SB 1223 DI 2 x 24 VDC/DQ 2 x 24 VDC, 200 кГц (6ES7223-3BD30-0XB0)	**SB 1223 DI 2 x 5 VDC / DQ 2 x 5 VDC, 200 кГц (6ES7223-3AD30-0XB0)**
	

① Поддерживает только входы-источники

② Для отдающих ток выходов подключить "Нагрузку" к "-" (показано на рис.). Для потребляющих ток выходов подключить "Нагрузку" к "+". [1] Поскольку как потребляющие, так и отдающие ток конфигурации поддерживаются той же самой схемой, активное состояние отдающей ток нагрузки противоположно активному состоянию потребляющей нагрузки. Отдающий ток выход проявляет положительную логику (Q бит и светодиод активны, когда течет ток в нагрузке), в то время как потребляющий ток выход проявляет отрицательную логику (Q бит и светодиод неактивны, когда есть ток в нагрузке). Если модуль вставлен без пользовательской программы, на выходе модуля по умолчанию будет 0 В, то есть потребляющая ток нагрузка будет включена.

Примечание

Проследить, чтобы провод соединения с корпусом был правильно заземлен. Потеря соединения с землей для высокоскоростных сигнальных плат цифрового вывода может вызвать ток утечки в объеме, достаточном для активации нагрузки постоянного тока. Если выходы используются для важных нагрузок постоянного тока, следует соблюдать особую осторожность, используя резервный заземляющий провод к SB.

Таблица A-203 Разводка контактов для SB 1223 DI 2 x 24 VDC/DQ 2 x 24 VDC, 200 кГц (6ES7223-3BD30-0XB0)

Контакт	**X19**
1	L+ / 24 В DC
2	M / 24 В DC
3	DI e.0
4	DI e.1

Контакт	X19
5	DQ e.0
6	DQ e.1

Таблица A-204 Разводка контактов для SB 1223 DI 2 x 5 VDC / DQ 2 x 5 VDC, 200 кГц (6ES7223-3AD30-0XB0)

Контакт	X19
1	L+ / 5 В DC
2	M / 5 В DC
3	DI e.0
4	DI e.1
5	DQ e.0
6	DQ e.1

A.13.4 Технические данные для SB 1223 с 2 x 24 В DC вход / 2 x 24 В DC выход

Таблица A-205 Общие технические данные

Технические данные	SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC
Заказной номер	6ES7223-0BD30-0XB0
Габаритные размеры (Ш x В x Г) (мм)	38 x 62 x 21
Вес	40 г
Рассеиваемая мощность	1,0 Вт
Потребляемый ток (SM шина)	50 мА
Потребляемый ток (24 В DC)	4 мА / вход

Таблица A-206 Цифровые входы

Технические данные	SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC
Количество входов	2
Тип	IEC Тур 1 потребитель
Номинальное напряжение	24 В DC при 4 мА, ном.
Длительно допустимое напряжение	До 30 В DC
Перенапряжение	35 В DC на 0,5 с
Сигнал логический 1 (мин.)	15 В DC при 2,5 мА
Сигнал логический 0 (макс.)	5 В DC при 1 мА
Скорость изменения входного сигнала высокоскоростных счетчиков (макс.)	Частота следования 1-фазных входных сигналов: 30 кГц (от 15 до 26 В DC) Частота следования 2-фазных входных сигналов: 20 кГц (от 15 до 26 В DC)

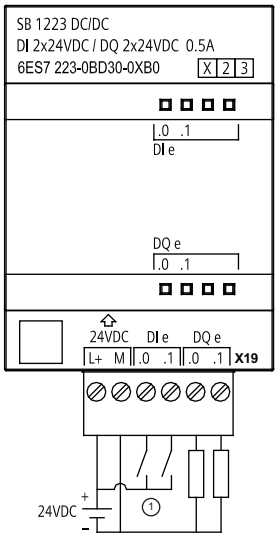
Технические данные		**SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC**
Электрическое разделение (между полевой стороной и логикой)		707 В DC (типовое испытание)
Изолированные группы		1
Время фильтрации	мкс установка:	0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0
	мс установка:	0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0
Количество одновременно активных входов		2
Длина кабеля (в метрах)		500 экранированный, 300 неэкранированный

Таблица A-207 Цифровые выходы

Технические данные	**SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC**
Выходы	2
Тип выхода	Транзисторные ключи MOSFET
Диапазон напряжений	от 20,4 до 28,8 В DC
Сигнал логический 1 при макс. токе	Мин. 20 В DC
Сигнал логический 0 при нагрузке 10 кОм	Макс. 0,1 В DC
Ток (макс.)	0,5 А
Ламповая нагрузка	5 Вт
Сопротивление активного выходного канала	Макс. 0,6 Ом
Ток утечки на выход	До 10 мкА
Частота следования выходных импульсов (PTO)	макс. 20 кГц, мин. 2 Гц [1]
Импульсный ток нагрузки, не более	5 А макс. в течение 100 мс
Защита от перегрузки	Нет
Электрическое разделение (между полевой стороной и логикой)	707 В DC (типовое испытание)
Изолированные группы	1
Ток на общий провод	1 А
Индуктивное фиксирующее напряжение	L+ - 48 В, 1 Вт рассеиваемая мощность
Задержка переключения	До 2 мкс из Выкл на Вкл До 10 мкс из Вкл на Выкл
Поведение при переходе из RUN в STOP	Последнее значение или замещающее значение (по умолчанию 0)
Количество одновременно активных выходов	2
Длина кабеля (в метрах)	500 м экранированный, 150 м неэкранированный

[1] В зависимости от используемого приемника импульсов и кабеля, дополнительный нагрузочный резистор (по крайней мере 10% от номинального тока) может улучшить качество импульсных сигналов и стойкость к воздействию помех.

Таблица A-208 Схема электрических соединений SB цифрового ввода/вывода

SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC (6ES7223-0BD30-0XB0)	
	

① Поддерживает только входы-потребители

Таблица A-209 Разводка контактов для SB 1223 DI с 2 x 24 В DC вход / 2 x 24 В DC выход (6ES7223-0BD30-0XB0)

Контакт	X19
1	L+ / 24 В DC
2	M / 24 В DC
3	DI e.0
4	DI e.1
5	DQ e.0
6	DQ e.1

A.14 Аналоговые сигнальные платы (SB)

A.14.1 Технические данные сигнальной платы SB 1231 с 1 аналоговым входом

Примечание

Для использования этой сигнальной платы потребуется CPU с прошивкой от версии V2.0.

Таблица A-210 Общие технические данные

Технические данные	**SB 1231 AI 1 x 12 бит**
Заказной номер	6ES7231-4HA30-0XB0
Габаритные размеры (Ш x В x Г) (мм)	38 x 62 x 21
Вес	35 г
Рассеиваемая мощность	0,4 Вт
Потребляемый ток (SM шина)	55 мА
Потребляемый ток (24 В DC)	Нет

Таблица A-211 Аналоговые входы

Технические данные	**SB 1231 AI 1x12 бит**
Количество входов	1
Тип	Напряжение или ток (дифференциальный)
Диапазон	±10 В, ±5 В, ±2,5 В или от 0 до 20 мА
Разрешение	11 бит + знаковый разряд
Цифровое представление полной шкалы (слово данных)	От -27648 до 27648
Выход за границы диапазона (слово данных)	Напряжение: От 32511 до 27649 / от -27649 до -32512 Ток: От 32511 до 27649 / от 0 до -4864 (см. Представление аналогового входа для напряжения и представление аналогового входа для тока (Страница 1594)).
Переполнения за пределами верхней/нижней границы диапазона (слово данных)	Напряжение: От 32767 до 32512 / от -32513 до -32768 Ток: От 32767 до 32512 / от -4865 до -32768 (см. Представление аналогового входа для напряжения и представление аналогового входа для тока (Страница 1594)).
Максимальное напряжение/ток	±35 В / ±40 мА
Сглаживание	Отсутствует, слабое, среднее или сильное (см. Реакция на скачок аналогового входа (Страница 1594)).
Подавление шумов	400, 60, 50 или 10 Гц (см. Время реакции аналогового входа для времени выборки (Страница 1594)).
Точность (25 °C / -20 до 60 °C)	±0,3% / ±0,6% по отношению к конечной точке шкалы
Полное входное сопротивление	Напряжение: 150 кОм; ток: 250 Ом
Принцип измерения	Преобразование фактического значения
Подавление синфазных сигналов	40 дБ, ном. значение при 60 Гц
Диапазон изменения входного напряжения	Суммарное значение напряжения сигнала и входного синфазного напряжения должно находиться в пределах от -35 В до +35 В
Электрическое разделение (между полевой стороной и логикой)	Нет
Длина кабеля (в метрах)	100 м, экранированная витая пара

Таблица A-212 Диагностика

Технические данные	SB 1231 AI 1 x 12 бит
Переполнения за пределами верхней/нижней границы диапазона	Да
Низкое напряжение 24 В DC	Нет

Измерительный преобразователь для SB 1231

Измерительные преобразователи доступны в 2-х и 4-х проводном исполнении, как показано ниже.

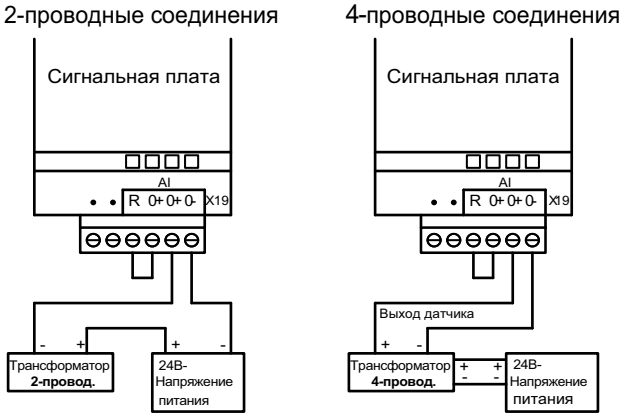


Таблица A-213 Схема электрических соединений для SB аналогового ввода

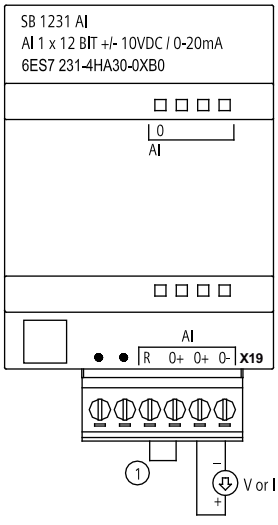
SB 1231 AI x 12 бит (6ES7231-4HA30-0XB0)	
	① Соединить "R" и "0+" для электрического тока. Примечание: Соединители должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.

Таблица A-214 Разводка контактов для SB 1231 AI x 12 бит (6ES7231-4HA30-0XB0)

Контакт	X19 (позолоченный)
1	Не подключено
2	Не подключено
3	AI R
4	AI 0+
5	AI 0+
6	AI 0-

A.14.2 Технические данные сигнальной платы SB 1232 с 1 аналоговым выходом

Таблица A-215 Общие технические данные

Технические данные	SB 1232 AQ 1 x 12 бит
Заказной номер	6ES7232-4HA30-0XB0
Габаритные размеры (Ш x В x Г) (мм)	38 x 62 x 21
Вес	40 г
Рассеиваемая мощность	1,5 Вт
Потребляемый ток (SM шина)	15 мА
Потребляемый ток (24 В DC)	40 мА (без нагрузки)

Таблица A-216 Аналоговые выходы

Технические данные	SB 1232 AQ 1 x 12 бит
Выходы	1
Тип	Напряжение или ток
Диапазон	±10 В или от 0 до 20 мА
Разрешение	Напряжение: 12 бит Ток: 11 бит
Цифровое представление полной шкалы (слово данных) См. раздел о диапазонах вывода для напряжения и тока (Страница 1596).	Напряжение: от -27.648 до 27.648; ток: Ток: от 0 до 27648
Точность (25 °C / -20 до 60 °C)	±0,5% / ±1% по отношению к конечной точке шкалы
Время установки (95% от нового значения)	Напряжение: 300 мкс (R), 750 мкс (1 мкФ) Ток: 600 мкс (1 мГн), 2 мс (10 мГн)
Полное сопротивление нагрузки	Напряжение: ≥ 1000 Ом Ток: ≤ 600 Ом
Поведение при переходе из RUN в STOP	Последнее значение или замещающее значение (по умолчанию 0)

Технические данные	SB 1232 AQ 1 x 12 бит
Электрическое разделение (между полевой стороной и логикой)	Нет
Длина кабеля (в метрах)	100 м, экранированная витая пара

Таблица A-217 Диагностика

Технические данные	SB 1232 AQ 1 x 12 бит
Переполнения за пределами верхней/нижней границы диапазона	Да
Замыкание на землю (только для режима по напряжению)	Да
Обрыв провода (только для режима по току)	Да

Таблица A-218 Схема электрических соединений для SB 1232 AO 1 x 12 бит

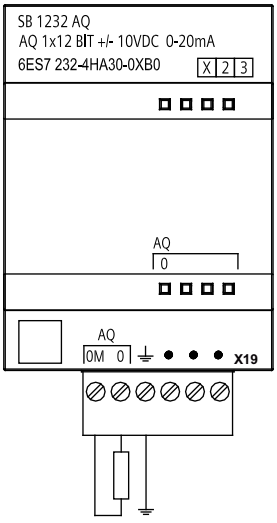
SB 1232 AQ 1 x 12 бит (6ES7232-4HA30-0XB0)	
	
Примечание: Соединители должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	

Таблица A-219 Разводка контактов для SB 1232 AQ 1 x 12 бит (6ES7232-4HA30-0XB0)

Контакт	X19 (позолоченный)
1	AQ 0M
2	AQ 0
3	Функциональное заземление
4	Не подключено
5	Не подключено
6	Не подключено

A.14.3 Диапазоны измерения аналоговых входов и выходов

A.14.3.1 Реакция на скачок аналоговых входов

Таблица A-220 Реакция на скачок (мс), от 0 до 10 В измеренная при 95%

Выбор сглаживания (усреднение по выборке)	**Выбор времени интегрирования**			
	400 Гц (2,5 мс)	**60 Гц (16,6 мс)**	**50 Гц (20 мс)**	**10 Гц (100 мс)**
Отсутствует (1 цикл): без усреднения	4,5 мс	18,7 мс	22,0 мс	102 мс
Слабое (4 цикла): 4 значения	10,6 мс	59,3 мс	70,8 мс	346 мс
Среднее (16 циклов): 16 значений	33,0 мс	208 мс	250 мс	1240 мс
Сильное (32 цикла): 32 значения	63,0 мс	408 мс	490 мс	2440 мс
Время выборки	**0,156 мс**	**1,042 мс**	**1,250 мс**	**6,250 мс**

A.14.3.2 Время выборки и время обновления для аналоговых входов

Таблица A-221 Время выборки и время обновления

Выбор	**Время выборки**	**Время обновления SB**
400 Гц (2,5 мс)	0,156 мс	0,156 мс
60 Гц (16,6 мс)	1,042 мс	1,042 мс
50 Гц (20 мс)	1,250 мс	1,25 мс
10 Гц (100 мс)	6,250 мс	6,25 мс

A.14.3.3 Диапазоны измерений аналоговых входов для напряжения и тока (SB и SM)

Таблица A-222 Представление аналогового входа для напряжения (SB и SM)

Система		**Диапазон измерения напряжения**				
Десятичная	**Шестнадцатеричная**	**±10 В**	**±5 В**	**±2,5 В**	**±1,25 В**	
32767	7FFF[1]	11,851 В	5,926 В	2,963 В	1,481 В	Выход за верхний установленный предел
32512	7F00					
32511	7EFF	11,759 В	5,879 В	2,940 В	1,470 В	Выход за границы диапазона
27649	6C01					

Система		**Диапазон измерения напряжения**				
Десятичная	**Шестнадцате-ричная**	**±10 В**	**±5 В**	**±2,5 В**	**±1,25 В**	
27648	6C00	10 В	5 В	2,5 В	1,250 В	Расчетный диапазон
20736	5100	7,5 В	3,75 В	1,875 В	0,938 В	
1	1	361,7 мкВ	180,8 мкВ	90,4 мкВ	45,2 мкВ	
0	0	0 В	0 В	0 В	0 В	
-1	FFFF					
-20736	AF00	-7,5 В	-3,75 В	-1,875 В	-0,938 В	
-27648	9400	-10 В	-5 В	-2,5 В	-1,250 В	
-27649	93FF					Выход за нижнюю границы диапазона
-32512	8100	-11,759 В	-5,879 В	-2,940 В	-1,470 В	
-32513	80FF					Выход сигнала за нижний установленный предел
-32768	8000	-11,851 В	-5,926 В	-2,963 В	-1,481 В	

[1] Возврат 7FFF возможен по одной из следующих причин: Выход за верхний установленный предел (см. таблицу), до получения правильных значений (например, сразу же при запуске) или при обнаружении обрыва провода.

Таблица A-223 Представление аналогового входа для тока (SB и SM)

Система		**Диапазон измерения тока**		
Десятичная	**Шестнадцате-ричная**	**от 0 до 20 мА**	**от 4 до 20 мА**	
32767	7FFF	> 23,52 мА	> 22,81 мА	Выход за верхний установленный предел
32511	7EFF	23,52 мА	22,81 мА	Выход за границы диапазона
27649	6C01			
27648	6C00	20 мА	20 мА	Ном. диапазон
20736	5100	15 мА	16 мА	
1	1	723,4 нА	4 мА + 578,7 нА	
0	0	0 мА	4 мА	
-1	FFFF			Выход за нижнюю границы диапазона
-4864	ED00	-3,52 мА	1,185 мА	
32767[1]	7FFF		< - 1,185 мА	Обрыв провода (от 4 до 20 мА)
-32768	8000	< -3,52 мА		Выход сигнала за нижний установленный предел (от 0 до 20 мА)

[1] Значение обрыва провода 32767 (16#7FFF) возвращается всегда, независимо от состояния предупреждения об обрыве провода.

A.14.3.4 Диапазоны измерений аналоговых выходов для напряжения и тока (SB и SM)

Таблица A-224 Представление аналогового выхода для напряжения (SB и SM)

Система		Диапазон выходного напряжения	
Десятичная	**Шестнадцатеричная**	**±10 В**	
32767	7FFF	См. Примечание 1	Выход за верхний установленный предел
32512	7F00	См. Примечание 1	
32511	7EFF	11,76 В	Выход за границы диапазона
27649	6C01		
27648	6C00	10 В	Расчетный диапазон
20736	5100	7,5 В	
1	1	361,7 мкВ	
0	0	0 В	
-1	FFFF	-361,7 мкВ	
-20736	AF00	-7,5 В	
-27648	9400	-10 В	
-27649	93FF		Выход за нижнюю границы диапазона
-32512	8100	-11,76 В	
-32513	80FF	См. Примечание 1	Выход сигнала за нижний установленный предел
-32768	8000	См. Примечание 1	

[1] При выходе за верхний или нижний установленный предел, аналоговые выходы используют замещающее значение для рабочего состояния STOP.

Таблица A-225 Представление аналогового выхода для тока (SB и SM)

Система		Диапазон выходного тока		
Десятичная	**Шестнадцатеричная**	**от 0 до 20 мА**	**от 4 до 20 мА**	
32767	7FFF	См. Примечание 1	См. Примечание 1	Выход за верхний установленный предел
32512	7F00	См. Примечание 1	См. Примечание 1	
32511	7EFF	23,52 мА	22,81 мА	Выход за границы диапазона
27649	6C01			
27648	6C00	20 мА	20 мА	Расчетный диапазон
20736	5100	15 мА	16 мА	
1	1	723,4 нА	4 мА + 578,7 нА	
0	0	0 мА	4 мА	
-1	FFFF		4 мА до 578,7 нА	Выход за нижнюю границы диапазона
-6912	E500		0 мА	
-6913	E4FF			Невозможно. Значение выходного сигнала ограничено до 0 мА.
-32512	8100			

Система		Диапазон выходного тока		
Десятичная	**Шестнадца-теричная**	**от 0 до 20 мА**	**от 4 до 20 мА**	
-32513	80FF	См. Примечание 1	См. Примечание 1	Выход сигнала за нижний установленный предел
-32768	8000	См. Примечание 1	См. Примечание 1	

[1] При выходе за верхний или нижний установленный предел, аналоговые выходы используют замещающее значение для рабочего состояния STOP.

A.14.4 Сигнальные платы для термопар (SB)

A.14.4.1 Технические данные сигнальной платы SB 1231 с 1 аналоговым входом для термопары

Примечание

Для использования этой сигнальной платы потребуется CPU с прошивкой от версии V2.0.

Таблица A-226 Общие технические данные

Технические данные	**SB 1231 AI 1 x 16 бит Thermocouple**
Заказной номер	6ES7231-5QA30-0XB0
Габаритные размеры (Ш x В x Г) (мм)	38 x 62 x 21
Вес	35 г
Рассеиваемая мощность	0,5 Вт
Потребляемый ток (SM шина)	5 мА
Потребляемый ток (24 В DC)	20 мА

Таблица A-227 Аналоговые входы

Технические данные	**SB 1231 AI 1x16 бит Thermocouple**
Количество входов	1
Тип	Плавающая термопара и мВ
Диапазон • Ном. диапазон (слово данных) • Переполнения за пределами верхней/нижней границы диапазона (слово данных) • Переполнения за пределами верхней/нижней границы диапазона (слово данных)	См. таблицу выбора фильтра для термопары (Страница 1599).

Технические данные		**SB 1231 AI 1x16 бит Thermocouple**
Разрешение	Температура	0,1 °C/0,1 °F
	Напряжение	15 бит плюс знаковый разряд
Максимальное входное напряжение		±35 В
Подавление шумов		85 дБ для выбранной настройки фильтра (10 Гц, 50 Гц, 60 Гц, 400 Гц)
Подавление синфазных сигналов		>120 дБ при 120 В AC
Полное сопротивление		≥ 10 МОм
Точность		См. Таблицу выбора термопар (Страница 1599).
Повторяемость		±0,05 % FS
Принцип измерения		Интегрирование
Время обновления модуля		См. таблицу выбора фильтра для термопары (Страница 1599).
Ошибка холодного спая		±1,5 °C
Электрическое разделение (между полевой стороной и логикой)		707 В DC (типовое испытание)
Длина кабеля (в метрах)		Макс. 100 м до датчика
Сопротивление линии		Макс. 100 Ом

Таблица A-228 Диагностика

Технические данные	**SB 1231 AI 1 x 16 бит Thermocouple**
Переполнения за пределами верхней/нижней границы диапазона[1]	Да
Обрыв провода[2, 3]	Да

[1] Диагностическая информация о переполнении будет передана в аналоговых значениях, даже если тревоги будут отключены в конфигурации модуля.

[2] Если тревога обрыва провода отключена, и имеет место обрыв в проводке датчика, модуль может возвращать случайные значения.

[3] Модуль выполняет контроль обрыва провода каждые 6 секунд, из-за чего время обновления каждые 6 секунд увеличивается на 9 мс для каждого активированного канала.

Аналоговый сигнальный модуль SM 1231 Thermocouple (TC) измеряет значение напряжения, поданного на входы модуля.

Аналоговая сигнальная плата SB 1231 Thermocouple (TC) измеряет значение напряжения, поданного на входы платы. Типом измерения температуры может быть "Термопара" или "Напряжение".

- "Термопара": Измеренное значение будет передано в градусах, умноженных на десять (например: 25,3 градуса будет представлено как десятичное число 253).
- "Напряжение": Номинальным предельным значением диапазона измерения будет десятичное 27648.

A.14.4.2 Принцип работы термопары

Термопары образуются, когда два разнородных металла электрически соединяются друг с другом. При соединении вырабатывается напряжение, которое пропорционально температуре в месте спая. Это напряжение является очень низким. Один микровольт может соответствовать нескольким градусам. Измерение напряжения термопары, компенсация дополнительных спаев и последующая линеаризация результатов являются основанием для измерения температуры с помощью термопар.

При подключении термопары к модулю SM 1231 Thermocouple, оба разнородных металлических провода подключаются к сигнальному соединению модуля. Место, где два разнородных провода соединены друг с другом, образует термопару датчика.

Еще две термопары образуются, когда два разнородных провода подключаются к сигнальному соединителю. Температура соединителя создает напряжение, которое добавляется к напряжению от термопары датчика. Если это напряжение не скорректировано, то передаваемая температура, будет отличаться от температуры датчика.

Компенсация холодного спая используется для корректировки влияния термопары соединителя. Таблицы для термопар основаны на опорной температуре соединения, обычно это нуль градусов по Цельсию. Компенсация холодного спая дает корректировку соединителю на ноль градусов по Цельсию. Компенсация холодного спая исправляет напряжение, измененное термопарой соединителя. Температура модуля измеряется внутри, и затем преобразуется в значение, которое будет добавлено к значению от датчика. Затем исправленное значение датчика линеаризуется, используя таблицы термопар.

Для оптимальной работы компенсации холодного спая модуль термопары должен быть расположен в среде со стабильной температурой. Медленное изменение (менее чем 0,1 °C/мин) температуры в месте нахождения модуля правильно компенсируется согласно техническим требованиям модуля. Движение воздуха через модуль также приводит к ошибкам компенсации холодного спая.

Для наилучшей компенсации ошибки холодного спая, можно использовать внешнюю изотермическую клеммную колодку. Модуль термопары поддерживает использование клеммной колодки с эталонной температурой 0 °C или 50 °C.

Таблицы выбора для SB 1231 Thermocouple (TC)

Диапазоны и точность для различных типов термопар, поддерживаемых сигнальной платой SB 1231 Thermocouple, представлены в таблице ниже.

Таблица A-229 Таблицу выбора термопар

Тип	Минимальная нижняя граница температуры[1]	Номинальная нижняя граница температуры	Номинальная верхняя граница температуры	Максимальная верхняя граница температуры[2]	Погрешность преобразования [3, 4] при 25 °C	Погрешность преобразования [1, 2] при -20 °C до 60 °C
J	-210,0 °C	-150,0 °C	1200,0 °C	1450,0 °C	±0,3 °C	±0,6 °C
	-346,0 °F	-238,0 °F	2192,0 °F	2642,0 °F	±0,5°F	±1,1 °F

Тип	Минимальная нижняя граница температуры[1]	Номинальная нижняя граница температуры	Номинальная верхняя граница температуры	Максимальная верхняя граница температуры[2]	Погрешность преобразования [3, 4] при 25 °C	Погрешность преобразования [1, 2] при -20 °C до 60 °C
K	-270,0 °C	-200,0 °C	1372,0 °C	1622,0 °C	±0,4 °C	±1,0 °C
	-454,0 °F	-328,0 °F	2501,6 °F	2951,6 °F	±0,7 °F	±1,8 °F
T	-270,0 °C	-200,0 °C	400,0 °C	540,0 °C	±0,5 °C	±1,0 °C
	-454,0 °F	-328,0 °F	752,0 °F	1004,0 °F	±0,9 °F	±1,8 °F
E	-270,0 °C	-200,0 °C	1000,0 °C	1200,0 °C	±0,3 °C	±0,6 °C
	-454,0 °F	-328,0 °F	1832,0 °F	2192,0 °F	±0,5°F	±1,1 °F
R & S	-50,0 °C	100,0 °C	1768,0 °C	2019,0 °C	±1,0 °C	±2,5 °C
	-58,0 °C	212,0 °F	3214,4 °F	3276,6 °F [5]	±1,8 °F	±4,5 °F
B	0,0 °C	200,0 °C	800,0 °C	--	±2,0 °C	±2,5 °C
	32,0 °F	392,0 °F	1472,0 °F	--	±3,6 °F	±4,5 °F
	--	800,0 °C	1820,0 °C	1820,0 °C	±1,0 °C	±2,3 °C
	--	1472,0 °F	3276,6 °F [5]	3276,6 °F [5]	±1,8 °F	±4,1 °F
N	-270,0 °C	-200,0 °C	1300,0 °C	1550,0 °C	±1,0 °C	±1,6 °C
	-454,0 °F	-328,0 °F	2372,0 °F	2822,0 °F	±1,8 °F	±2,9 °F
C	0,0 °C	100,0 °C	2315,0 °C	2500,0 °C	±0,7 °C	±2,7 °C
	32,0 °F	212,0 °F	3276,6 °F [5]	3276,6 °F [5]	±1,3 °F	±4,9 °F
TXK/XK(L)	-200,0 °C	-150,0 °C	800,0 °C	1050,0 °C	±0,6 °C	±1,2 °C
	-328,0 °F	302,0 °F	1472,0 °F	1922,0 °F	±1,1 °F	±2,2 °F
Напряжение	-32512	-27648 -80 мВ	27648 80 мВ	32511	±0,05 %	±0,1 %

[1] Значения от термопары ниже минимальной нижней границы температуры возвращаются как -32768.

[2] Значения от термопары выше максимальной верхней границы температуры возвращаются как 32767.

[3] Погрешность внутренней температурной компенсации холодного спая составляет ±1,5% для всех диапазонов измерения температуры. Эта погрешность добавляется к приведенным в таблице погрешностям. Для перехода в штатный режим модуль должен отработать не менее 30 минут.

[4] При наличии помех в диапазоне частот от 970 до 990 МГц точность преобразования SM 1231 AI 4 x 16 бит TC может снижаться.

[5] Нижнее предельное значение 3276,6 с информацией в °F

Таблица A-230 Таблицы выбора фильтра для SB 1231 Thermocouple (TC)

Частота подавления помех (Гц)	Время интегрирования (мс)	Время обновления сигнальной платы (сек)
10	100	0.306
50	20	0.066
60	16.67	0.056
400[1]	10	0.036

[1] Для того, чтобы достичь заявленной точности и разрешения для модуля при фильтре 400 Гц, время интегрирования составляет 10 мс. Этот фильтр также отсекает помехи с частотой 100 Гц и 200 Гц.

Для измерительных термопар рекомендуется использовать время интегрирования в 100 мс. Использование более низких значений времени интегрирования увеличивает погрешность повторяемости при измерении температуры.

Примечание

После включения модуль выполняет внутреннюю калибровку аналого-цифрового преобразователя. В течение этого времени модуль возвращает значение 32767 в каждом канале, пока действительные данные не станут доступны на этом канале. При необходимости следует учесть это время инициализации в программе пользователя.

Таблица A-231 Схема электрических соединений SB 1231 AI 1 x 16 Thermocouple

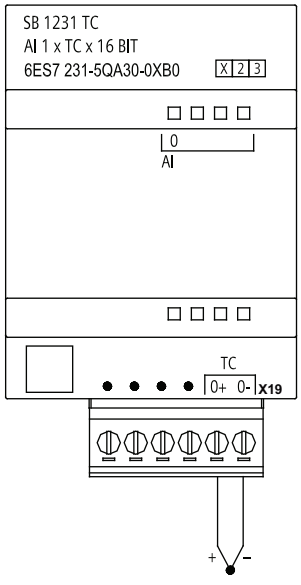
SB 1231 AI 1 x 16 бит Thermocouple (6ES7231-5QA30-0XB0)	
	
Примечание: Соединители должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.	

Таблица A-232 Разводка контактов для SB 1231 AI 1 x 16 бит Thermocouple (6ES7231-5QA30-0XB0)

Контакт	X19 (позолоченный)
1	Не подключено
2	Не подключено
3	Не подключено
4	Не подключено
5	AI 0- /TC
6	AI 0+ /TC

A.14.5 Сигнальные платы RTD (SB)

A.14.5.1 Технические данные сигнальной платы SB 1231 с 1 аналоговым входом RTD

Примечание

Для использования этой сигнальной платы потребуется CPU с прошивкой от версии V2.0.

Таблица A-233 Общие технические данные

Технические данные	SB 1231 AI 1 x 16 бит RTD
Заказной номер	6ES7231-5PA30-0XB0
Габаритные размеры (Ш x В x Г) (мм)	38 x 62 x 2
Вес	35 г
Рассеиваемая мощность	0,7 Вт
Потребляемый ток (SM шина)	5 мА
Потребляемый ток (24 В DC)	25 мА

Таблица A-234 Аналоговые входы

Технические данные	SB 1231 AI 1 x 16 бит RTD
Количество входов	1
Тип	Рекомендованные для модуля термометры сопротивления и сопротивления

Технические данные		**SB 1231 AI 1 x 16 бит RTD**
Диапазон • Ном. диапазон (слово данных) • Переполнения за пределами верхней/нижней границы диапазона (слово данных) • Переполнения за пределами верхней/нижней границы диапазона (слово данных)		См. Таблицы выбора (Страница 1605).
Разрешение	Температура	0,1 °C / 0,1 °F
	Напряжение	15 бит плюс знаковый разряд
Максимальное входное напряжение		±35 В
Подавление шумов		85 дБ (10 Гц, 50 Гц, 60 Гц, 400 Гц)
Подавление синфазных сигналов		>120 дБ
Полное сопротивление		≥ 10 МОм
Точность		См. Таблицы выбора (Страница 1605).
Повторяемость		±0,05 % FS
Потери мощности датчика, не более		0,5 мВт
Принцип измерения		Интегрирование
Время обновления модуля		См. Таблицы выбора (Страница 1605).
Электрическое разделение (между полевой стороной и логикой)		707 В DC (типовое испытание)
Длина кабеля (в метрах)		Макс. 100 м до датчика
Сопротивление линии		20 Ом, 2,7 Ом для 10 Ом термометров сопротивления макс.

Таблица A-235 Диагностика

Технические данные	**SB 1231 AI 1 x 16 бит RTD**
Переполнения за пределами верхней/нижней границы диапазона[1, 2]	Да
Обрыв провода[3]	Да

1 Диагностическая информация о переполнении будет передана в аналоговых значениях, даже если тревоги будут отключены в конфигурации модуля.

2 Для диапазонов измерения сопротивления проверка на переполнения за пределами нижней границы диапазона никогда не активируется.

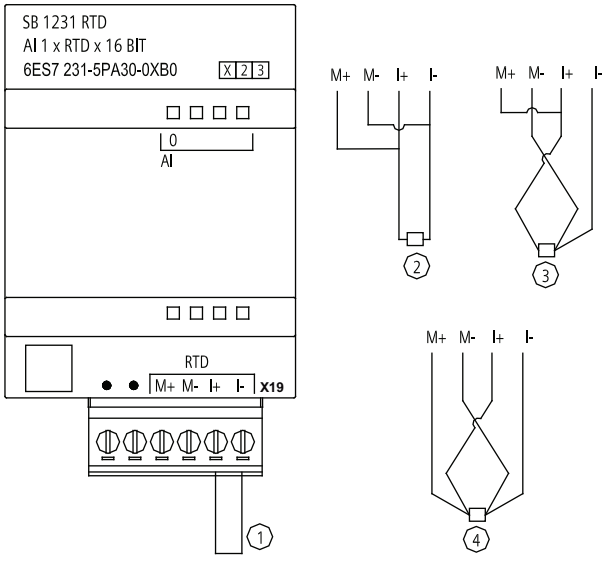
3 Если тревога обрыва провода отключена, и имеет место обрыв в проводке датчика, модуль может возвращать случайные значения.

Аналоговая сигнальная плата SB 1231 Thermocouple RTD измеряет значение сопротивления, подключенного ко входам платы. В качестве тип измерения можно выбрать "Сопротивление" или "Тепловое сопротивление".

- "Сопротивление": Номинальным предельным значением диапазона измерения будет десятичное 27648.
- "Тепловое сопротивление": Измеренное значение будет передано в градусах, умноженных на десять (например: 25,3 градуса будет представлено как десятичное число 253). Значения климатического диапазона будут переданы в градусах, умноженных на сто (например: 25,34 градуса будет представлено как десятичное число 2534).

Сигнальная плата SB 1231 RTD поддерживает измерения с помощью 2-проводного, 3-проводного и 4-проводного подключения датчика.

Таблица A-236 Схема электрических соединений SB 1231 AI 1 x 16 бит RTD

SB 1231 AI 1 x 16 бит RTD (6ES7231-5PA30-0XB0)


① Закольцевать не занятые входы RTD

② 2проводный RTD

③ 3проводный RTD

④ 4проводный RTD

Примечание: Соединители должны быть позолоченными. См. "Приложение C, запасные части" для заказного номера.

Таблица A-237 Разводка контактов для SB 1231 AI 1 x 16 бит RTD (6ES7231-5PA30-0XB0)

Контакт	X19 (позолоченный)
1	Не подключено
2	Не подключено
3	AI 0 M+ /RTD
4	AI 0 M- /RTD
5	AI 0 I+ /RTD
6	AI 0 I- /RTD

A.14.5.2 Таблицы выбора для SB 1231 RTD

Таблица A-238 Диапазоны и точность для различных датчиков, поддерживаемых модулем RTD

Температурный коэффициент	Тип RTD	Минимальная нижняя граница температуры[1]	Номинальная нижняя граница температуры	Номинальная верхняя граница температуры	Максимальная верхняя граница температуры[2]	Погрешность преобразования при 25 °C	Погрешность преобразования при -20 °C до 60 °C
Pt 0,003850	Pt 100 climatic	-145,00 °C	-120,00 °C	-145,00 °C	-155,00 °C	±0,20 °C	±0,40 °C
ITS90	Pt 10	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±1,0 °C	±2,0 °C
DIN EN 60751	Pt 50	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±0,5 °C	±1,0 °C
	Pt 100						
	Pt 200						
	Pt 500						
	Pt 1000						
Pt 0,003902	Pt 100	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±0,5 °C	±1,0 °C
Pt 0,003916	Pt 200						
Pt 0,003920	Pt 500						
	Pt 1000						
Pt 0,003910	Pt 10	-273,2 °C	-240,0 °C	1100,0 °C	1295 °C	±1,0 °C	±2,0 °C
	Pt 50	-273,2 °C	-240,0 °C	1100,0 °C	1295 °C	±0,8 °C	±1,6 °C
	Pt 100						
	Pt 500						
Ni 0,006720	Ni 100	-105,0 °C	-60,0 °C	250,0 °C	295,0 °C	±0,5 °C	±1,0 °C
Ni 0,006180	Ni 120						
	Ni 200						
	Ni 500						
	Ni 1000						
LG-Ni 0,005000	LG-Ni 1000	-105,0 °C	-60,0 °C	250,0 °C	295,0 °C	±0,5 °C	±1,0 °C
Ni 0,006170	Ni 100	-105,0 °C	-60,0 °C	180,0 °C	212,4 °C	±0,5 °C	±1,0 °C
Cu 0,004270	Cu 10	-240,0 °C	-200,0 °C	260,0 °C	312,0 °C	±1.0 °	±2,0 °C

Температурный коэффициент	Тип RTD	Минимальная нижняя граница температуры[1]	Номинальная нижняя граница температуры	Номинальная верхняя граница температуры	Максимальная верхняя граница температуры[2]	Погрешность преобразования при 25 °C	Погрешность преобразования при -20 °C до 60 °C
Cu 0,004260	Cu 10	-60,0 °C	-50,0 °C	200,0 °C	240,0 °C	±1,0 °C	±2,0 °C
	Cu 50	-60,0 °C	-50,0 °C	200,0 °C	240,0 °C	±0,6 °C	±1,2 °C
	Cu 100						
Cu 0,004280	Cu 10	-240,0 °C	-200,0 °C	200,0 °C	240,0 °C	±1,0 °C	±2,0 °C
	Cu 50	-240,0 °C	-200,0 °C	200,0 °C	240,0 °C	±0,7 °C	±1,4 °C
	Cu 100						

[1] Значения от RTD ниже минимальной нижней границы температуры возвращаются как -32768.

[2] Значения от RTD выше максимальной верхней границы температуры возвращаются как +32768.

Таблица A-239 Стойкость к:

Диапазон	Минимальная нижняя граница сопротивления	Номинальная нижняя граница сопротивления	Номинальная верхняя граница сопротивления	Максимальная верхняя граница сопротивления[1]	Погрешность преобразования при 25 °C	Погрешность преобразования при -20 °C до 60 °C
150 Ом	Неприменимо	0 (0 Ом)	27648 (150 Ом)	176,383 Ом	±0.05%	±0.1%
300 Ом	Неприменимо	0 (0 Ом)	27648 (300 Ом)	352,767 Ом	±0.05%	±0.1%
600 Ом	Неприменимо	0 (0 Ом)	27648 (600 Ом)	705,534 Ом	±0.05%	±0.1%

[1] Значения сопротивления выше максимальной верхней границы сопротивления возвращаются как 32767.

Примечание

Модуль возвращает 32767 для любого активированного канала без подключенного датчика. Если также разрешено обнаружение обрыва провода, модуль активирует соответствующие красные светодиоды.

Для 10 Ом диапазонов тепловых сопротивлений оптимальная точность будет достигнута при использовании 4-проводного подключения.

Сопротивление соединительных проводов при 2-проводном подключении вызовет ошибку интерпретации показаний датчика. В этом случае точность измерения не гарантируется.

Таблица A-240 Подавление помех и время обновления для модулей RTD

Выбор частоты подавления помех	Время интегрирования	1-канальный 4/2-проводный модуль Время обновления (сек)	1-канальный 3-проводный модуль Время обновления (сек)
400 Гц (2,5 мс)	10 мс [1]	0.036	0.071
60 Гц (16,6 мс)	16,67 мс	0.056	0.111
50 Гц (20 мс)	20 мс	0.066	1.086
10 Гц (100 мс)	100 мс	0.306	0.611

[1] Для того, чтобы достичь заявленной точности и разрешения для модуля при фильтре 400 Гц, время интегрирования составляет 10 мс. Этот фильтр также отсекает помехи с частотой 100 Гц и 200 Гц.

Примечание

После включения модуль выполняет внутреннюю калибровку аналого-цифрового преобразователя. В течение этого времени модуль возвращает значение 32767 в каждом канале, пока действительные данные не станут доступны на этом канале. При необходимости следует учесть это время инициализации в программе пользователя. Поскольку конфигурация модуля может изменить продолжительность времени инициализации, следует проверить поведение модуля в конкретной конфигурации. При необходимости можно включать логику в свою программу пользователя, чтобы подстроиться под время инициализации модуля.

A.15 Плата буферной батареи BB 1297

Плата буферной батареи BB 1297

Плата буферной батареи S7-1200 BB 1297 (батарейная плата) служит для долговременной буферизации часов реального времени. Она может быть установлена в слот для сигнальных плат S7-1200 CPU (от версии прошивки 3.0). Необходимо включить BB 1297 в конфигурацию устройства и загрузить конфигурацию оборудования в CPU, чтобы батарейная плата выполняла свои функции.

Батарея (тип CR1025) не входит в комплект поставки BB 1297 и должна приобретаться отдельно.

Примечание

Механическая конструкция BB 1297 разработана для использования с CPU с версией прошивки 3.0 и выше.

Не использовать BB 1297 с более ранними версиями CPU, поскольку соединитель BB 1297 не подходит к CPU.

⚠ ПРЕДУПРЕЖДЕНИЕ

Установка не специфицированной батареи в плату BB 1297, или подключение не специфицированной батареи к цепи любым другим способом могут привести к возгоранию или повреждению компонентов, а также непредсказуемой работе оборудования.

Такое непредсказуемое поведение может стать причиной летального исхода, тяжких травм персонала и/или материального ущерба.

Для буферизации часов реального времени следует использовать только специфицированную батарею CR1025.

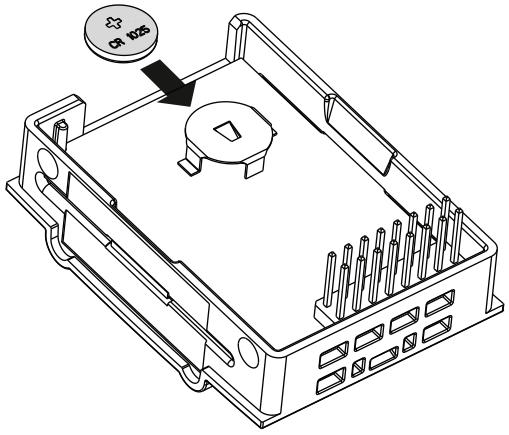
Таблица A-241 Общие технические данные

Технические данные	Плата буферной батареи BB 1297
Заказной номер	6ES7297-0AX30-0XA0
Габаритные размеры (Ш x В x Г) (мм)	38 x 62 x 21
Вес	28 г
Рассеиваемая мощность	0,5 Вт
Потребляемый ток (SM шина)	11 мА
Потребляемый ток (24 В DC)	Нет

Батарея (не включена в поставку)	Плата буферной батареи BB 1297
Буферизация отключения сети	Около 1 года
Тип батареи	CR1025 см. Установка или замена батареи в плате буферной батареи BB 1297 (Страница 62)
Номинальное напряжение	3 В
Номинальная емкость	Минимум 30 мАчас

Диагностика	Плата буферной батареи BB 1297
Критический уровень заряда батареи	< 2,5 В
Диагностика батареи	Индикатор низкого уровня заряда батареи: • Ровное свечение индикатора MAINT на фронтальной панели CPU янтарным цветом при низком уровне напряжения. • Диагностическое сообщение: 16#06:2700 "Требуется обслуживание субмодуля: По меньшей мере одна батарея разряжена (BATTF)"
Состояние батареи	Бит состояния батареи 0 = батарея в порядке 1 = низкий уровень напряжения
Обновление информации о состоянии батареи	При включении питания, затем раз в день при работе CPU.

Таблица A-242 Схема сборки для батарейной платы BB 1297

BB 1297 батарейная плата (6ES7297-0AX30-0XA0)	
	

A.16 Коммуникационные интерфейсы

A.16.1 PROFIBUS

A.16.1.1 CM 1242-5 ВЕДОМОЕ УСТРОЙСТВО PROFIBUS DP

Таблица A-243 Технические данные CM 1242-5

Технические данные	
Заказной номер	6GK7242-5DX30-0XE0
Интерфейсы	
Подключение к PROFIBUS	9-штырьковый гнездовой D-sub соединитель
Максимальное потребление тока на интерфейсе PROFIBUS при подключении сетевых компонентов (например, оптических сетевых компонентов)	15 мА при 5 В (только для оконечной нагрузки шины) *)
Допустимые условия окружающей среды	
Температура окружающей среды	
• при хранении	• от -40 °C до 70 °C
• при транспортировке	• от -40 °C до 70 °C
• при работе с вертикальной установкой (DIN-рейка горизонтально)	• от 0 °C до +55°C
• при работе с горизонтальной установкой (DIN-рейка вертикально)	• от 0 °C до 45 °C
Относительная влажность воздуха при 25 °C во время работы, без образования конденсата, макс.	95 %

Технические данные	
Степень защиты	IP20
Питающее напряжение, потребляемый ток и мощность потерь	
Тип питающее напряжение	DC
Питание от внутренней шины	5 В
Потребляемый ток (тип.)	150 мА
Эффективная рассеиваемая мощность (тип.)	0,75 Вт
Гальваническая развязка • между интерфейсом PROFIBUS и массой • между интерфейсом PROFIBUS и внутренними цепями	710 В AC на 1 минуту
Габаритные размеры и вес	
• Ширина • Высота • Глубина	• 30 мм • 100 мм • 75 мм
Вес • Вес-нетто • Вес с упаковкой	 • 115 г • 152 г

*)Токовая нагрузка от внешнего потребителя, подключенного между VP (контакт 6) и DGND (контакт 5), не должна превышать максимум 15 мА (защита от короткого замыкания) для оконечной нагрузки шины.

A.16.1.2 Разводка контактов для соединителя D-sub модуля CM 1242-5

Интерфейс PROFIBUS

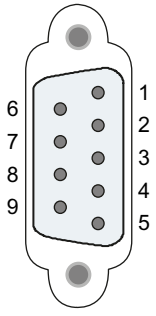


Таблица A-244 Разводка контактов для соединителя D-sub

Контакт	Описание	Контакт	Описание
1	- не используется -	6	P5V2: Питающее напряжение +5В
2	- не используется -	7	- не используется -
3	RxD/TxD-P: Линия данных B	8	RxD/TxD-N: Линия данных A

Контакт	Описание	Контакт	Описание
4	RTS	9	- не используется -
5	M5V2: Опорный потенциал канала данных (земля DGND)	Корпус	Клемма заземления

A.16.1.3 СМ 1243-5 ВЕДУЩЕЕ УСТРОЙСТВО PROFIBUS DP

Таблица A-245 Технические данные СМ 1243-5

Технические данные	
Заказной номер	6GK7243-5DX30-0XE0
Интерфейсы	
Подключение к PROFIBUS	9-штырьковый гнездовой D-sub соединитель
Максимальное потребление тока на интерфейсе PROFIBUS при подключении сетевых компонентов (например, оптических сетевых компонентов)	15 мА при 5 В (только для оконечной нагрузки шины) *)
Допустимые условия окружающей среды	
Температура окружающей среды	
• при хранении	• от -40 °C до 70 °C
• при транспортировке	• от -40 °C до 70 °C
• при работе с вертикальной установкой (DIN-рейка горизонтально)	• от 0 °C до +55°C
• при работе с горизонтальной установкой (DIN-рейка вертикально)	• от 0 °C до 45 °C
Относительная влажность воздуха при 25 °C во время работы, без образования конденсата, макс.	95 %
Степень защиты	IP20
Питающее напряжение, потребляемый ток и мощность потерь	
Тип питающее напряжение	DC
Питающее напряжение / внешнее	24 В
• минимум	• 19,2 В
• максимум	• 28,8 В
Потребляемый ток (тип.)	
• из 24 В DC	• 100 мА
• от внутренней шины S7-1200	• 0 мА
Эффективная рассеиваемая мощность (тип.)	
• из 24 В DC	• 2,4 Вт
• от внутренней шины S7-1200	• 0 Вт
Внешнее питающее напряжение 24 В DC	
• Мин. сечение кабеля	• мин.: 0,14 мм^2 (AWG 25)
• Макс. сечение кабеля	• макс.: 1,5 мм^2 (AWG 15)
• Момент затяжки зажимов под винт	• 0,45 Нм (4 фунта на дюйм)

Технические данные	
Гальваническая развязка • между интерфейсом PROFIBUS и массой • между интерфейсом PROFIBUS и внутренними цепями	710 В AC на 1 минуту
Габаритные размеры и вес	
• Ширина • Высота • Глубина	• 30 мм • 100 мм • 75 мм
Вес • Вес-нетто • Вес с упаковкой	 • 134 г • 171 г

*)Токовая нагрузка от внешнего потребителя, подключенного между VP (контакт 6) и DGND (контакт 5), не должна превышать максимум 15 мА (защита от короткого замыкания) для оконечной нагрузки шины.

Примечание

CM 1243-5 (модуль ведущего устройства PROFIBUS) должен получать питание от источника 24 В DC для питания датчиков CPU.

A.16.1.4 Разводка контактов для соединителя D-sub модуля CM 1243-5

Интерфейс PROFIBUS

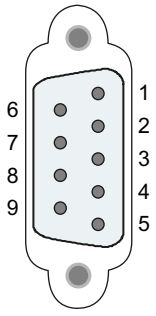


Таблица A-246 Разводка контактов для соединителя D-sub

Контакт	Описание	Контакт	Описание
1	- не используется -	6	VP: питание +5 В только для терминальных резисторов шины; не для питания внешних устройств
2	- не используется -	7	- не используется -

Контакт	Описание	Контакт	Описание
3	RxD/TxD-P: Линия данных B	8	RxD/TxD-N: Линия данных A
4	CNTR-P: RTS	9	- не используется -
5	DGND: Земля для сигналов данных и VP	Корпус	Клемма заземления

Кабель PROFIBUS

Примечание

Заземление экрана кабеля PROFIBUS

Экран кабеля PROFIBUS должен быть заземлен.

Для этого зачистить изоляцию на одном из концов кабеля PROFIBUS и соединить экран с функциональным заземлением.

A.16.2 CP 1242-7

Примечание

Модуль CP 1242-7 GPRS не имеет морского сертификата

У модуля CP 1242-7 GPRS нет морского сертификата.

Примечание

Для использования этих модулей потребуется CPU с прошивкой от версии V2.0.

A.16.2.1 CP 1242-7 GPRS

Таблица A-247 Технические данные CP 1242-7 GPRS V2

Технические данные	
Заказной номер	6GK7242-7KX3-0XE0
Беспроводный интерфейс	
Подключение антенны	SMA гнездо
Полное сопротивление	50 Ом
Беспроводная технология	
Мощность передатчика, не более	• GSM 850, класс 4: +33 дБм ±2дБм • GSM 900, класс 4: +33 ДБм ±2ДБм • GSM 1800, класс 1: +30 ДБм ±2ДБм • GSM 1900, класс 1: +30 ДБм ±2ДБм

Технические данные	
GPRS	Мультислотовый класс 10 Класс прибора B Схема кодирования 1...4 (GMSK)
SMS	Режим вывода: MO Сервис: точка-к-точке
Допустимые условия окружающей среды	
Температура окружающей среды	
• при хранении	• от -40 °C до 70 °C
• при транспортировке	• от -40 °C до 70 °C
• при работе с вертикальной установкой (DIN-рейка горизонтально)	• от 0 °C до +55°C
• при работе с горизонтальной установкой (DIN-рейка вертикально)	• от 0 °C до 45 °C
Относительная влажность воздуха при 25 °C во время работы, без образования конденсата, макс.	95 %
Степень защиты	IP20
Питающее напряжение, потребляемый ток и мощность потерь	
Тип питающее напряжение	DC
Питающее напряжение / внешнее	24 В
• минимум	• 19,2 В
• максимум	• 28,8 В
Потребляемый ток (тип.)	
• из 24 В DC	• 100 мА
• от внутренней шины S7-1200	• 0 мА
Эффективная рассеиваемая мощность (тип.)	
• из 24 В DC	• 2,4 Вт
• от внутренней шины S7-1200	• 0 Вт
Электропитание 24 В DC	
• Мин. сечение кабеля	• мин.: 0,14 мм^2 (AWG 25)
• Макс. сечение кабеля	• макс.: 1,5 мм^2 (AWG 15)
• Момент затяжки зажимов под винт	• 0,45 Нм (4 фунта на дюйм)
Гальваническая развязка Между цепью питания и внутренней электроникой	710 В AC на 1 минуту
Габаритные размеры и вес	
• Ширина	• 30 мм
• Высота	• 100 мм
• Глубина	• 75 мм
Вес	
• Вес-нетто	• 133 г
• Вес с упаковкой	• 170 г

Примечание

Предотвращение нарушений в работе CPU из-за антенны

Возможны нарушения в работе CPU, если антенна расположена слишком близко или если была выбрана не рекомендуемая антенна. Рекомендуемые антенны можно найти в кратком руководстве по эксплуатации для антенны ANT794-4MR для LTE / UMTS / GSM (https://support.industry.siemens.com/cs/ww/en/view/23119005) (доступно только на английском и немецком языках).

A.16.2.2 GSM/GPRS антенна ANT794-4MR

Технические данные GSM/GPRS антенны ANT794-4MR

ANT794-4MR	
Заказной номер	6NH9860-1AA00
Мобильные сети	GSM/GPRS
Диапазоны частот	• от 824 до 960 МГц (GSM 850, 900) • от 1710 до 1880 МГц (GSM 1800, 900) • от 1900 до 2200 МГц (GSM / UMTS)
Тип антенны	круговая
Коэффициент усиления	0 дБ
Полное сопротивление	50 Ом
Коэффициент стоячей волны (SWR)	< 2,0
Макс. мощность	20 Вт
Полярность	линейная, вертикальная
Соединитель	SMA
Длина антенного кабеля	5 м
Наружный материал корпуса	Поливинилхлорид, стойкий к воздействию ультрафиолетового излучения
Степень защиты	IP20
Допустимые условия окружающей среды • Рабочая температура • Транспортировка и хранение • Относительная влажность воздуха	 • от -40 °C до +70 °C • от -40 °C до +70 °C • 100 %
Наружный материал корпуса	Поливинилхлорид, стойкий к воздействию ультрафиолетового излучения
Комплект поставки	Антенна с антенным кабелем длиной 5 м, оснащенным штекером SMA
Габариты (Г x В) в мм	25 x 193
Вес • Антенна вместе с кабелем • Крепеж	 • 310 g • 54 g
Монтаж	С помощью прилагаемой скобы

A.16.2.3 Плоская антенна ANT794-3M

Технические данные плоской антенны ANT794-3M

ANT794-3M		
Заказной номер	6NH9870-1AA00	
Мобильные сети	**GSM 900**	**GSM 1800/1900**
Диапазоны частот	890 - 960 МГц	1710 - 1990 МГц
Коэффициент стоячей волны (VSWR)	≤ 2:1	≤ 1,5:1
Потери возврата (Tx)	≈ 10 дБ	≈ 14 дБ
Коэффициент усиления	0 дБ	
Полное сопротивление	50 Ом	
Макс. мощность	10 Вт	
Антенный кабель	HF кабель RG 174 со штекером SMA	
Длина кабеля	1,2 м	
Степень защиты	IP64	
Допустимый диапазон температур	от -40 °C до +75°C	
Стойкость к воздействию огня	UL 94 V2	
Наружный материал корпуса	Лак ABS PA-765, светло серый (RAL 7035)	
Габариты (Г x В) в мм	70,5 x 146,5 x 20,5	
Вес	130 г	

A.16.3 Коммуникационный модуль ведущего устройства AS-Interface CM 1243-2

A.16.3.1 Технические данные ведущего AS-i устройства CM 1243-2

Таблица A-248 Технические данные ведущего AS-i устройства CM 1243-2

Технические данные	
Заказной номер	3RK7243-2AA30-0XB0
Версия прошивки	V1.0
Дата	01.12.2011
Интерфейсы	
Макс. потребляемый ток от внутренней шины S7-1200 через AS-i кабель	 Макс. 250 мА , Напряжение питания коммуникационной шины S7-1200 = 5 В DC Макс. 100 мА
Максимально допустимая нагрузка по току между контактами ASI+/ASI-	8 A

Технические данные	
Разводка контактов	См. раздел Электрическое подключение ведущего устройства AS-i шины (Страница 1617)
Сечение проводника	0,2 мм² (AWG 24) ... 3,3 мм² (AWG 12)
Момент затяжки ASI соединителя	0,56 Нм
Допустимые условия окружающей среды	
Температура окружающей среды	
при хранении	от -40 °C до 70 °C
при транспортировке	от -40 °C до 70 °C
при работе с вертикальной установкой (DIN-рейка горизонтально)	от 0 °C до 55 °C
при работе с горизонтальной установкой (DIN-рейка вертикально)	от 0 °C до 45 °C
Относительная влажность воздуха при 25 °C во время работы, без образования конденсата, макс.	95 %
Степень защиты	IP20
Питающее напряжение, потребляемый ток и мощность потерь	
Тип питающее напряжение	DC
Потребляемый ток (тип.)	
от внутренней шины S7-1200	200 мА
Суммарное снижение мощности (тип.):	
• от внутренней шины S7-1200	1 Вт
• через AS-i кабель	2,4 Вт
Габаритные размеры и вес	
Ширина	30 мм
Высота	100 мм
Глубина	75 мм
Вес	
Вес-нетто	122 г
Вес с упаковкой	159 г

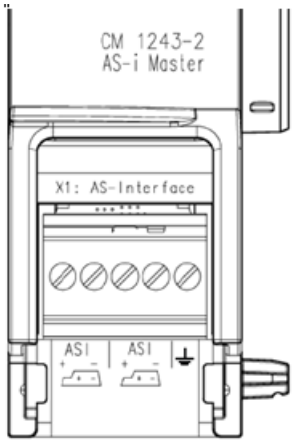
A.16.3.2 Электрическое подключение ведущего устройства AS-i шины

Источник питания ведущего устройства AS-i шины CM 1243-2

Ведущее устройство AS-i шины CM 1243-2 получает питание от коммуникационной шины S7-1200. Это означает, что диагностическое сообщение может быть послано в S7-1200 и после сбоя напряжения питания AS-i шины. Соединение с коммуникационной шиной находится на ведущем устройстве AS-i шины CM 1243-2 справа.

Терминальный блок AS-Interface

Съемный терминальный блок для подключения AS-i кабеля расположен за нижней крышкой на передней стороне ведущего устройства AS-i шины CM 1243-2.



Если используется AS-i профилированный кабель, то можно узнать правильную полярность с помощью символа



Информацию о том, как демонтировать и повторно установить терминальный блок, можно найти в главе "Монтаж" (Страница 67).

Примечание

Максимальная нагрузка на контакты

Максимальная нагрузка соединительных контактов составляет максимум 8 А. Если эта значение превышено в AS-i кабеле, то ведущее устройство AS-i шины CM 1243-2 не должно быть "закольцовано" через AS-i кабель, а должно подключаться через ответвление (только одна соединительная пара, подведенная к ведущему устройству AS-i шины CM 1243-2).
Кроме этого следует убедиться, что используемые кабели пригодны для рабочих температур не менее 75°C, если ток проводится через ведущее устройство AS-i шины и присутствуют токи выше 4 ампер.
Дополнительную информацию о подключении AS-i кабеля можно найти в разделе "Монтаж, подключение и ввод в эксплуатацию модулей" в руководстве "Ведущее устройство AS-i шины CM 1243-2 и разделительный модуль DCM 1271 для SIMATIC S7-1200".

Разводка контактов

Метка	Значение
ASI+	Подключение к AS-i – положительная полярность
ASI–	Подключение к AS-i – отрицательная полярность
	Функциональное заземление

A.16.4 RS232, RS422 и RS485

A.16.4.1 Технические данные платы CB 1241 RS485

Примечание

Для использования этой платы потребуется CPU с прошивкой от версии V2.0.

Таблица A-249 Общие технические данные

Технические данные	CB 1241 RS485
Заказной номер	6ES7241-1CH30-1XB0
Габаритные размеры (Ш x В x Г) (мм)	38 x 62 x 21
Вес	40 г

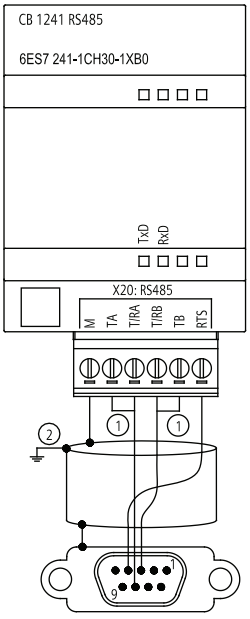
Таблица A-250 Передатчик и приемник

Технические данные	CB 1241 RS485
Тип	RS 485 (2-проводный полудуплекс)
Диапазон изменения синфазных сигналов	-7 ... +12 В в течение 1 секунды, синусоидальное напряжение 3 В длительно
Дифференциальное выходное напряжение передатчика	мин. 2 В при R_L = 100 Ом мин. 1,5 В при R_L = 54 Ом
Терминальные сопротивления и сопротивления смещения	10 кОм по отношению к +5 В для сигнала B (контакт 3 на RS485) 10 кОм по отношению к земле (GND) для сигнала A (контакт 4 на RS485)
Дополнительная оконечная нагрузка	Замкнуть контакт TB на T/RB, эффективный импеданс нагрузки составляет 127 Ом, подключение к RS485 контакт 3 Замкнуть контакт TA на T/RA, эффективный импеданс нагрузки составляет 127 Ом, подключение к RS485 контакт 4
Входное сопротивление приемника	Не менее 5,4 кОм, включая терминальное сопротивление
Порог чувствительности приемника	мин. ±0,2 В, типовой гистерезис 60 мВ
Гальваническая развязка Между сигналом RS485 и массой Между сигналом RS485 и логикой CPU	707 В DC (типовое испытание)
Длина кабеля (в метрах)	макс. 1000 м.
Скорость передачи	300 бод, 600 бод, 1,2 кбит/с, 2,4 кбит/с, 4,8 кбит/с, 9,6 кбит/с (по умолчанию), 19,2 кбит/с, 38,4 кбит/с, 57,6 кбит/с, 76,8 кбит/с, 115,2 кбит/с.
Четность	Без контроля четности (по умолчанию), совпадение при контроле по чётности, совпадение при контроле по нечётности, контроль по единичному биту (бит четности всегда установлен в 1), контроль по нулевому биту (бит четности всегда установлен в 0).

Технические данные	**CB 1241 RS485**
Количество стоповых битов	1 (по умолчанию), 2
Управление потоком	Не поддерживается
Время ожидания	от 0 до 65535 мс

Таблица A-251 Электропитание

Технические данные	**CB 1241 RS485**
Мощность потерь	1,5 Вт
Потребляемый ток (SM шина), макс.	50 мА
Макс. потребляемый ток (24 В DC)	80 мА

CB 1241 RS485 (6ES7241-1CH30-1XB0)	
	
① Подсоединить "TA" и TB" как показано на рисунке, чтобы включить оконечную нагрузку. (Включать оконечную нагрузку только на оконечных устройствах RS485 сети.)	
② Использовать экранированную витую пару и соединить экран кабеля с корпусом.	

Оконечная нагрузка включается только для двух концов сети RS485. Оконечная нагрузка не включается для устройств между обоими оконечными устройствами.

Дополнительную информацию можно найти здесь "Включение оконечной нагрузки в шинном разъеме RS485" (Страница 1118).

Таблица A-252 Разводка контактов для СВ 1241 RS485 (6ES7241-1CH30-1XB0)

Контакт	**9-контактный разъем**	**X20**
1	RS485 / логическая земля	--
2	RS485 / не используется	--
3	RS485 / TxD+	4 - T/RB
4	RS485 / RTS	6 - RTS
5	RS485 / логическая земля	--
6	RS485 / питание 5 В	--
7	RS485 / не используется	--
8	RS485 / TxD-	3 - T/RA
9	RS485 / не используется	--
Корпус		1 - M

A.16.4.2 Технические данные модуля CM 1241 RS232

Таблица A-253 Общие технические данные

Технические данные	**CM 1241 RS232**
Заказной номер	6ES7241-1AH32-0XB0
Размеры (мм)	30 x 100 x 75
Вес	150 г

Таблица A-254 Передатчик и приемник

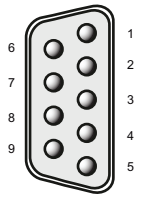
Технические данные	**CM 1241 RS232**
Тип	RS232 (дуплекс)
Выходное напряжение передатчика	мин. +/- 5 В при R_L = 3 кОм
Выходное напряжение передатчика	макс. +/- 15 В DC
Входное сопротивление приемника	мин. 3 кОм
Порог чувствительности приемника	мин. 0,8 В низкий уровень, макс. 2,4 В высокий уровень, типовой гистерезис 0,5 В
Входное напряжение приемника	макс. +/- 30 В DC
Гальваническая развязка Между сигналом RS-232 и массой Между сигналом RS-232 и логикой CPU	707 В DC (типовое испытание)
Длина кабеля (в метрах)	макс. 10 м.

Технические данные	CM 1241 RS232
Скорость передачи	300 бод, 600 бод, 1,2 кбит/с, 2,4 кбит/с, 4,8 кбит/с, 9,6 кбит/с (по умолчанию), 19,2 кбит/с, 38,4 кбит/с, 57,6 кбит/с, 76,8 кбит/с, 115,2 кбит/с.
Четность	Без контроля четности (по умолчанию), совпадение при контроле по чётности, совпадение при контроле по нечётности, контроль по единичному биту (бит четности всегда установлен в 1), контроль по нулевому биту (бит четности всегда установлен в 0).
Количество стоповых битов	1 (по умолчанию), 2
Управление потоком	аппаратное, программное
Время ожидания	от 0 до 65535 мс

Таблица A-255 Электропитание

Технические данные	CM 1241 RS232
Мощность потерь	1 Вт
Из +5 В DC	200 мА

Таблица A-256 RS232 соединитель (штекер)

Контакт	Описание	Соединитель (штекер)	Контакт	Описание
1 DCD	Обнаружение носителя данных: Вход		6 DSR	Набор данных готов: Вход
2 RxD	Прием данных от DCE: Вход		7 RTS	Запрос на передачу: Выход
3 TxD	Передача данных в DCE: Выход		8 CTS	Готовность к передаче: Вход
4 DTR	Готовность терминала данных: Выход		9 RI	Кольцевой индикатор (не используется)
5 GND	Логическая земля		SHELL	Корпусная земля

A.16.4.3 Технические данные модуля CM 1241 RS422/485

Технические данные модуля CM 1241 RS422/485

Таблица A-257 Общие технические данные

Технические данные	CM 1241 RS422/485
Заказной номер	6ES7241-1CH32-0XB0
Габаритные размеры (Ш x В x Г) (мм)	30 x 100 x 75
Вес	155 г

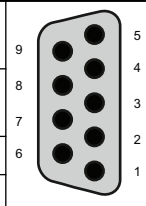
Таблица A-258 Передатчик и приемник

Технические данные	**CM 1241 RS422/485**
Тип	RS422 или RS485, 9-штырьковый гнездовой D-sub соединитель
Диапазон изменения синфазных сигналов	-7 ... +12 В в течение 1 секунды, синусоидальное напряжение 3 В длительно
Дифференциальное выходное напряжение передатчика	мин. 2 В при R_L = 100 Ом мин. 1,5 В при R_L = 54 Ом
Терминальные сопротивления и сопротивления смещения	10 кОм по отношению к +5 В для сигнала B (контакт 3 на PROFIBUS) 10 кОм по отношению к земле (GND) для сигнала A (контакт 8 на PROFIBUS) Предлагаются внутренние опции смещения или без внутреннего смещения. Во всех случаях требуется внешний терминальный резистор, см. Включение оконечной нагрузки в шинном разъеме RS485 (Страница 1118) и Конфигурирование RS422 и RS485 в системном руководстве по программируемому контроллеру S7-1200 (Страница 1171).
Входное сопротивление приемника	Не менее 5,4 кОм, включая терминальное сопротивление
Порог чувствительности приемника	мин. ±0,2 В, типовой гистерезис 60 мВ
Гальваническая развязка Между сигналом RS485 и массой Между сигналом RS485 и логикой CPU	707 В DC (типовое испытание)
Длина кабеля (в метрах)	макс. 1000 м (в зависимости от скорости передачи)
Скорость передачи	300 бод, 600 бод, 1,2 кбит/с, 2,4 кбит/с, 4,8 кбит/с, 9,6 кбит/с (по умолчанию), 19,2 кбит/с, 38,4 кбит/с, 57,6 кбит/с, 76,8 кбит/с, 115,2 кбит/с.
Четность	Без контроля четности (по умолчанию), совпадение при контроле по чётности, совпадение при контроле по нечётности, контроль по единичному биту (бит четности всегда установлен в 1), контроль по нулевому биту (бит четности всегда установлен в 0).
Количество стоповых битов	1 (по умолчанию), 2
Управление потоком	XON/XOFF поддерживается для режима RS422
Время ожидания	от 0 до 65535 мс

Таблица A-259 Электропитание

Технические данные	**CM 1241 RS422/485**
Мощность потерь	1,1 Вт
Из +5 В DC	220 мА

Таблица A-260 RS485 или RS422 соединитель (гнездо)

Контакт	Описание	Соединитель (гнездо)	Контакт	Описание
1	Логическая или коммуникационная земля		6 PWR	+5 В последовательно с резистором 100 Ом: Выход
2 TxD+ [1]	Подключено для RS422 Не используется для RS485: Выход		7	Не подключено
3 TxD+ [2]	Сигнал B (RxD/TxD+): Вход/выход		8 TXD- [2]	Сигнал A (RxD/TxD-): Вход/выход
4 RTS [3]	Запрос на передачу/RTS (TTL уровень): Выход		9 TXD- [1]	Подключено для RS422 Не используется для RS485: Выход
5 GND	Логическая или коммуникационная земля		SHELL	Корпусная земля

[1] Контакт 2 (TxD+) и контакт 9 (TxD-) - это сигналы передачи RS422.

[2] Контакт 3 (RxD/Tx+) и контакт 8 (RxD/TxD-) - это сигналы передачи и приема RS485. Для RS422 контакт 3 соответствует сигналу RxD+, а контакт 8 - сигналу RxD-.

[3] RTS - это сигнал TTL уровня для управления другими полудуплексными приборами, поддерживающими этот сигнал. Находится в активном состоянии при обмене данными и в пассивном состоянии во всех остальных случаях.

A.17 Телесервис (TS адаптер и модульный TS адаптер)

Следующие руководства/справочники содержат технические характеристики TS адаптера IE Basic и модульного TS адаптера:

- Промышленные инструменты программирования
  Модульный TS адаптер
- Промышленные инструменты программирования
  TS адаптер IE Basic

Для получения дополнительной информации и документации по продукту, обратиться к веб-сайту каталога продукции для TS адаптера (https://eb.automation.siemens.com/mall/en/de/Catalog/Search?searchTerm=TS%20Adapter%20IE%20basic&tab=).

A.18 Карты памяти SIMATIC

Емкость	Заказной номер
32 ГБ	6ES7954-8LT02-0AA0
2 ГБ	6ES7954-8LP01-0AA0
256 МБ	6ES7954-8LL02-0AA0
24 МБ	6ES7954-8LF02-0AA0
12 МБ	6ES7954-8LE02-0AA0
4 МБ	6ES7954-8LC02-0AA0

A.19 Имитаторы входных сигналов

Таблица A-261 Общие технические данные

Технические данные	8-канальный имитатор	14-канальный имитатор	CPU 1217C имитатор
Заказной номер	6ES7274-1XF30-0XA0	6ES7274-1XH30-0XA0	6ES7274-1XK30-0XA0
Габаритные размеры (Ш x В x Г) (мм)	43 x 35 x 23	67 x 35 x 23	93 x 40 x 23
Вес	20 г	30 г	43 г
Входы и выходы	8	14	14
Использование с	CPU 1211C, CPU 1212C	CPU 1214C, CPU 1215C	CPU 1217C

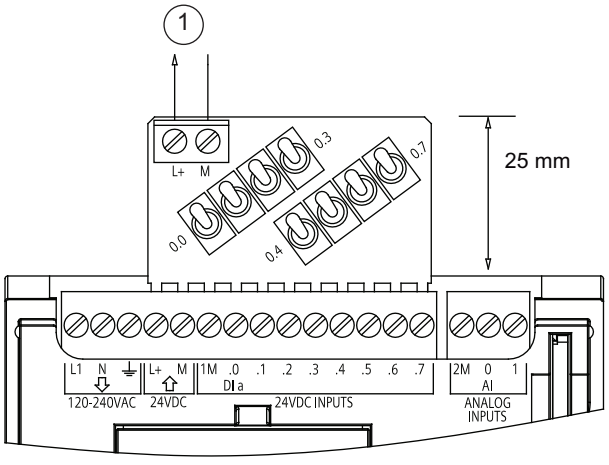
⚠ ПРЕДУПРЕЖДЕНИЕ

Безопасное использование имитаторов входных сигналов

Эти имитаторы входных сигналов не одобрены для использования во взрывоопасных производственных помещениях класса I, раздела 2, и класса I, зоны 2. Переключатели представляют потенциальную опасность возникновения искры / взрыва, если используются в помещениях класса I, раздела 2 или класса I, зоны 2. Несанкционированное использование может привести к смерти или тяжелым травмам персонала и/или повреждению оборудования.

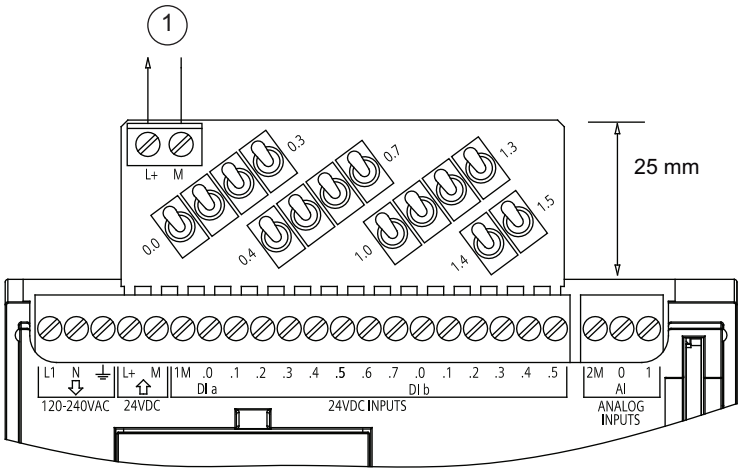
Следует использовать эти имитаторы входных сигналов только в безопасных помещениях. Не использовать их в помещениях класса I, раздела 2, и класса I, зоны 2.

8-канальный имитатор (6ES7274-1XF30-0XA0)



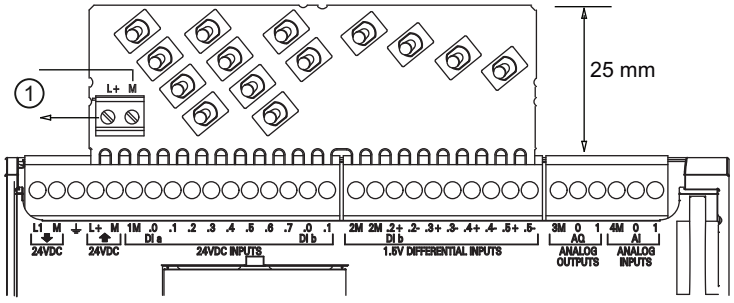
① Питание датчиков 24 В DC

14-канальный имитатор (6ES7274-1XF30-0XA0)



① Питание датчиков 24 В DC

CPU 1217C имитатор (6ES7274-1XK30-0XA0)



① Питание датчиков 24 В DC

A.20 Модуль потенциометров для S7-1200

Модуль потенциометров S7-1200 является дополнительным аксессуаром для S7-1200 CPU. Каждый потенциометр создает выходное напряжение, пропорциональное положению потенциометра, чтобы управлять каждым из двух аналоговых входов CPU с 0 до 10 В DC. Для установки потенциометра:

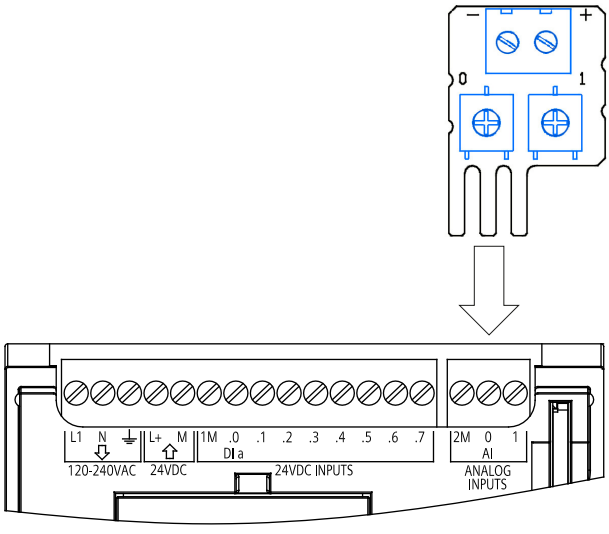
1. Вставить 'пальцы' печатной платы в любой терминальный блок аналогового входа S7-1200 CPU и подключить внешний источник питания постоянного тока к 2-позиционному соединителю на модуле потенциометров.
2. Использовать небольшую отвертку для настройки потенциометров: повернуть потенциометр по часовой стрелке (вправо), чтобы увеличить выходное напряжение, и против часовой стрелки (влево), чтобы уменьшить выходное напряжение.

Примечание

Придерживаться мер по защите элементов, подверженных опасности разрушения в результате электростатического заряда, при обращении с модулем потенциометров S7-1200.

Технические данные	**Модуль потенциометров для S7-1200**
Заказной номер	6ES7274-1XA30-0XA0
Использование с	Любые S71200 CPU
Количество потенциометров:	2
Габаритные размеры (Ш x В x Г) (мм)	20 x 33 x 14
Вес	26 г
Пользовательский вход по напряжению на 2-позиционном соединителе[1] (класс 2, ограниченное питание или питание датчиков от PLC)	от 16,4 В DC до 28,8 В DC
Длина кабеля (в метрах)/тип	< 30 м, экранированная витая пара
Входной потребляемый ток	макс. 10 мА
Выходное напряжение потенциометра на аналоговые входы S7-1200 CPU[1]	От 0 В DC до 10,5 В DC
Гальваническая развязка	Без электрического разделения
Температура окружающей среды	от -20 °C до +60 °C

[1] Стабильность выходного напряжения модуля потенциометров зависит от качества подаваемого пользователем на 2-позиционный соединитель входного напряжения - следует рассматривать его как напряжение аналогового входа.

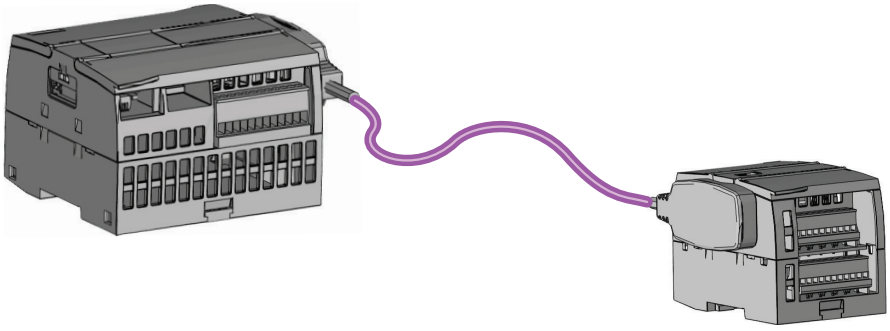


A.21 Кабель для модулей расширения

Таблица A-262 Расширительный кабель

Технические данные	
Заказной номер	6ES7290-6AA30-0XA0
Длина кабеля	2 м
Вес	200 г

Дополнительную информацию о монтаже и демонтаже расширительного кабеля S7-1200 можно найти в разделе "Монтаж" (Страница 68).



A.22 Сопутствующие продукты

A.22.1 Блок питания PM 1207

PM 1207 - это блок питания для SIMATIC S7-1200. Блок имеет следующие характеристики:

- Вход 120/230 В AC, выход 24 В DC/2,5 А

Для получения дополнительной информации и документации по продукту, обратиться к веб-сайту каталога продукции для PM 1207 (https://mall.industry.siemens.com/mall/en/de/Catalog/Product/6EP1332-1SH71).

A.22.2 Компактный коммутационный модуль CSM 1277

Модуль CSM1277 - это коммутатор Industrial Ethernet. С его помощью можно увеличить число интерфейсов Ethernet для организации одновременного обмена данными между S7-1200 и программатором, приборами и системами человеко-машинного интерфейса, другими системами автоматизации. Блок имеет следующие характеристики:

- 4 гнезда RJ45 для подключения к Industrial Ethernet
- съемный 3-полюсный терминальный блок с контактами под винт для подключения цепи питания 24 В DC сверху

- Светодиоды индикации состояний и диагностики коммуникационных портов Industrial Ethernet
- Заказной номер 6GK7277-1AA00-0AA0

Для получения дополнительной информации и документации по продукту, обратиться к веб-сайту каталога продукции для CSM 1277 (https://eb.automation.siemens.com/mall/en/de/Catalog/Search?searchTerm=csm%201277&tab=).

A.22.3 Коммуникационный модуль CM CANopen

Коммуникационный модуль CM CANopen является сменным модулем между SIMATIC S7-1200 PLC и любым устройством, использующим CANopen. Коммуникационный модуль CM CANopen может быть сконфигурирован как ведущее или как ведомое устройство. Существует два коммуникационных модуля CM CANopen modules: модуль CANopen (заказной номер 021620-B) и модуль для повышенной нагрузки CANopen (Ruggedized) (заказной номер 021730-B).

Модуль CANopen имеет следующие характеристики:

- Возможность подключения до 3 модулей на CPU
- Подключение до 16 ведомых узлов CANopen
- Передача 256 байт входных и 256 байт выходных данных для каждого модуля
- 3 диагностических светодиода для индикации состояний модуля, сети и каналов ввода-вывода
- Возможность сохранения конфигурации сети CANopen в PLC
- Модуль интегрируется в каталог оборудования конфигурационных средств TIA Portal
- Конфигурирование CANopen посредством CANopen Configuration Studio (в комплекте) или через любой другой внешний инструмент конфигурирования CANopen
- Соответствует профилям соединения CANopen CiA 301 ревизии 4.2 и CiA 302 ревизии 4.1.
- Поддержка прозрачной передачи данных CAN 2.0A для реализации нестандартных протоколов обмена данными
- Наличие готовых функциональных блоков для программирования контроллера в TIA Portal
- В объем поставки модулей CM CANopen включены: разъём D-sub с клеммами под винт для подсетей, CM CANopen Configuration Studio CD и конфигурационный USB-кабель.

Для получения дополнительной информации и документации по продукту, обратиться к веб-сайту каталога продукции для CM CANopen.

A.22.4 Коммуникационный модуль RF120C

Коммуникационный модуль RF120C обеспечивает прямое и простое подключение предлагаемых Siemens считывателей данных систем идентификации RFID и систем считывания кодов к S7-1200. Считыватель подключается к RF120C посредством соединения "точка-точка". До трех коммуникационных модулей могут быть подключены к S7-1200 слева от CPU. Коммуникационный модуль RF120C конфигурируется через TIA Portal. Заказной номер для коммуникационного модуля RF120C - 6GT2002-0LA00.

Для получения дополнительной информации и документации по продукту, обратиться к веб-сайту каталога продукции для RF120C.

A.22.5 SM 1238 Energy Meter

Модуль измерения параметров электроэнергии SM 1238 Energy Meter 480 В AC предназначен для использования в системе S7-1200. Он регистрирует свыше 200 различных параметров сети переменного тока. Это обеспечивает учет требовании к энергии для отдельных компонентов производственного оборудования уже на уровне производственных машин. Используя измеренные значения, предоставленные энергометром SM 1238, можно определить энергопотребление и рассчитать потребность в энергии.

Для получения дополнительной информации и документации по продукту и технических данных, обратиться к веб-сайту каталога продукции для SM 1238 Energy Meter (https://support.industry.siemens.com/cs/ww/en/view/109483435).

A.22.6 Электронные системы взвешивания SIWAREX

SIWAREX WP231, WP241 и WP251

В S7-1200 могут использоваться электронные системы взвешивания SIWAREX WP231, WP241 и WP251. Эти модули поддерживают все функции современной системы автоматизации, как то встроенная коммуникация, оперативное управление и мониторинг, диагностика, а также инструментарий конфигурирования в TIA Portal.

- Модуль взвешивания SIWAREX WP231 (https://support.industry.siemens.com/cs/us/en/view/90229056) с калибровкой (1 канал) для тензометрических весовых ячеек (1-4 MB/B) для SIMATIC S7-1200, интерфейс RS-485 и Ethernet, встроенные I/O: 4 DI / 4 DQ, 1 AQ (0/4...20 MA)
- Модуль взвешивания SIWAREX WP241 (https://support.industry.siemens.com/cs/ww/en/view/90229063) для использования с SIMATIC S7-1200 на движущихся ленточных конвейерах (1 канал) для тензометрических весовых ячеек (1-4 MB/B), интерфейс RS-485 и Ethernet, встроенные I/O: 4 DI / 4 DQ, 1 AQ (0/4...20 MA)
- Модуль взвешивания SIWAREX WP251 для автоматизации процессов дозирования и наполнения (1 канал) для тензометрических весовых ячеек (1-4 MB/B) с SIMATIC S7-1200, интерфейс RS-485 и Ethernet, встроенные I/O: 4 DI / 4 DQ, 1 AQ (0/4...20 MA)

См. также

SIWAREX WP251 (https://support.industry.siemens.com/cs/ww/en/view/109481751)

Расчет баланса мощностей

B

В CPU имеется внутренний источник питания, который, наряду с CPU, обеспечивает энергией модули расширения и другие нагрузки 24 В DC.

Существует четыре типа модулей расширения:

- Сигнальные модули (SM) подключаются с правой стороны от CPU. Каждый CPU позволяет подключить определенное максимальное количество сигнальных модулей безотносительно к балансу мощности.
  - К CPU 1214C, CPU 1215C и CPU 1217C может быть подключено 8 сигнальных модулей.
  - К CPU 1212C может быть подключено 2 сигнальных модуля.
  - К CPU 1211C не могут быть подключены сигнальные модули.
- Коммуникационные модули (CM) подключаются с левой стороны от CPU. К каждому CPU может быть подключено до 3 коммуникационных модулей, независимо от баланса мощностей.
- Сигнальные платы (SB), коммуникационные платы (CB) и платы буферной батареи (BB) подключаются сверху на CPU. К каждому CPU может быть подключена 1 сигнальная плата, коммуникационная плата ии плата буферной батареи.

С помощью следующей информации можно рассчитать, какую мощность модуль CPU может предоставить для конкретной конфигурации.

Каждый CPU предоставляет постоянное напряжение 5 В и 24 В:

- 5 В постоянного тока для возможно подключенных модулей расширения. Если модули расширения потребляют больше постоянного напряжения 5 В, чем может предоставить внутренний источник питания, следует отказаться от модулей расширения, пока баланс мощностей CPU снова не восстановится.
- У каждого CPU есть питание датчиков 24 В DC для подачи напряжения постоянного тока 24 В на встроенные входы и релейные катушки модулей расширения. Если CPU потребляют больше постоянного напряжения 24 В, чем может предоставить внутренний источник питания, то можно подключить дополнительный внешний источник питания 24 В для модулей расширения. Источник питания 24 В подключается вручную к входам и релейным катушкам.

⚠ ПРЕДУПРЕЖДЕНИЕ

Подключение внешнего источника питания 24 В DC параллельно источнику питания датчиков 24 В DC может привести к конфликту между двумя источниками, так как каждый из них стремится установить свой собственный уровень выходного напряжения.

Результатом этого конфликта может быть сокращение срока службы или немедленный выход из строя одного или обоих источников питания. Непредсказуемое поведение системы может привести к тяжким телесным повреждениям и/или материальному ущербу.

Блок питания датчиков постоянного тока в CPU и внешний источник питания должны подключаться к различным нагрузкам. При этом разрешается только одно соединение между обоими источниками напряжения.

Некоторые входные порты 24 В системы PLC соединены друг с другом, при этом общий логический нулевой провод соединяет между собой несколько M клемм. Вход источника питания 24 В DC на CPU, вход питания катушек на SM и не имеющий потенциальной развязки вход питания аналогового модуля являются примерами цепей, которые соединяются друг с другом, если в технических данных они обозначены, как не имеющие потенциальной развязки. Все не изолированные M клеммы должны быть подсоединены к одному и тому же внешнему опорному потенциалу.

⚠ ПРЕДУПРЕЖДЕНИЕ

Подключение не изолированных клемм M к различным опорным потенциалам вызовет появление непредусмотренных токов, которые могут привести к повреждению или к непредсказуемому поведению целевой системы и подключенного оборудования.

Такое повреждение или непредсказуемое поведение системы в свою очередь может стать причиной тяжких телесных повреждений и/или материального ущерба.

Все не изолированные M клеммы в системе PLC всегда должны подключаться к одному и тому же опорному потенциалу.

Информация о балансах мощности CPU и потребляемой мощности сигнальных модулей представлена в технических данных (Страница 1433).

Примечание

Превышение баланса мощности CPU может привести к невозможности подключения, максимального количества модулей, допустимого для конкретного CPU.

Пример для баланса мощности

В следующем примере показан пример расчета баланса мощности для конфигурации с одним CPU 1214C AC/DC/RLY, одной SB 1223 с 2 входами 24 В DC и 2 выходами 24 В DC, одним CM 1241, тремя SM 1223 с 8 DC входами / 8 релейными выходами и одним 8-канальным входным SM 1221 постоянного тока. В этом примере представлено в общей сложности 48 входов и 36 выходов.

Примечание

CPU уже выделил необходимую для внутренних катушек реле мощность. Потребляемая внутренними катушками реле мощность может не включаться в расчет баланса мощности.

В этом примере CPU предоставляет достаточно напряжения 5 В DC для сигнальных модулей, но напряжения питание датчиков 24 В DC недостаточно для всех входов и дополнительных реле. Входам и выходам требуется 456 мА, а CPU предоставляет только 400 мА. Для питания всех входов и выходов 24 В DC в этой конфигурации не хватает как минимум 56 мА.

Таблица B-1 Пример для баланса мощности

Баланс мощностей CPU	**5 В DC**	**24 В DC**
CPU 1214C AC/DC/RLY	1600 мА	400 мА
Минус		
Системные требования	**5 В DC**	**24 В DC**
CPU 1214C, 14 входов	-	14 * 4 мА = 56 мА
1 шт. SB 1223 с 2 x 24 В DC вход / 2 x 24 В DC выход	50 мА	2 * 4 мА = 8 мА
1 шт. CM 1241 RS422/485, 5 В	220 мА	
3 шт. SM 1223, напряжение 5 В	3 * 145 мА = 435 мА	-
1 шт. SM 1221, напряжение 5 В	1 * 105 мА = 105 мА	-
3 шт. SM 1223, по 8 входов на каждом	-	3 * 8 * 4 мА = 96 мА
3 шт. SM 1223, по 8 релейных выходов на каждом	-	3 * 8 * 11 мА = 264 мА
1 шт. SM 1221 с 8 входами	-	8 * 4 мА = 32 мА
Общая потребляемая мощность	810 мА	456 мА
Равно		
Баланс мощностей	**5 В DC**	**24 В DC**
Общий баланс мощностей	790 мА	(56 мА)

Форма для вычисления баланса мощности

С помощью этой таблицы можно рассчитать, какую мощность S7-1200 CPU может предоставить для конкретной конфигурации. В технических данных (Страница 1433) можно найти информацию о доступной мощности для отдельных моделей CPU и о потребляемой мощности используемых сигнальных модулей.

Таблица B-2 Расчет баланса мощностей

Баланс мощностей CPU	**5 В DC**	**24 В DC**
Минус		
Системные требования	**5 В DC**	**24 В DC**
Общая потребляемая мощность		
Равно		
Баланс мощностей	**5 В DC**	**24 В DC**
Общий баланс мощностей		

Информация для оформления заказа

C

C.1 Модули CPU

Таблица C-1 S71200 CPU

Варианты CPU		Заказной номер
CPU 1211C	CPU 1211C DC/DC/DC	6ES7211-1AE40-0XB0
	CPU 1211C AC/DC/RLY	6ES7211-1BE40-0XB0
	CPU 1211C DC/DC/RLY	6ES7211-1HE40-0XB0
CPU 1212C	CPU 1212C DC/DC/DC	6ES7212-1AE40-0XB0
	CPU 1212C AC/DC/RLY	6ES7212-1BE40-0XB0
	CPU 1212C DC/DC/RLY	6ES7212-1HE40-0XB0
CPU 1214C	CPU 1214C DC/DC/DC	6ES7214-1AG40-0XB0
	CPU 1214C AC/DC/RLY	6ES7214-1BG40-0XB0
	CPU 1214C DC/DC/RLY	6ES7214-1HG40-0XB0
CPU 1215C	CPU 1215C DC/DC/DC	6ES7215-1AG40-0XB0
	CPU 1215C AC/DC/RLY	6ES7215-1BG40-0XB0
	CPU 1215C DC/DC/RLY	6ES7215-1HG40-0XB0
CPU 1217C	CPU 1217C DC/DC/DC	6ES7217-1AG40-0XB0

C.2 Сигнальные модули (SM), сигнальные платы (SB) и батарейные платы (BB)

Таблица C-2 Сигнальные модули (SM)

Сигнальные модули		Заказной номер
Цифровой ввод	SM 1221 8 x вход 24 В DC (потребитель/источник)	6ES7221-1BF32-0XB0
	SM 1221 16 x вход 24 В DC (потребитель/источник)	6ES7221-1BH32-0XB0
Цифровой вывод	SM 1222 8 x выход 24 В DC (источник)	6ES7222-1BF32-0XB0
	SM 1222 16 x выход 24 В DC (источник)	6ES7222-1BH32-0XB0
	SM 1222 16 x выход 24 В DC (потребитель)	6ES7222-1BH32-1XB0
	SM 1222 8 x релейный выход	6ES7222-1HF32-0XB0
	SM 1222 8 x релейный выход (переключение)	6ES7222-1XF32-0XB0
	SM 1222 16 x релейный выход	6ES7222-1HH32-0XB0

Сигнальные модули		Заказной номер
Цифровой ввод/вывод	SB 1223 с 8 x 24 В DC вход (потребитель/источник) / 8 x 24 В DC выход (источник)	6ES7223-1BH32-0XB0
	SM 1223 с 16 x 24 В DC вход (потребитель/источник) / 16 x 24 В DC выход (источник)	6ES7223-1BL32-0XB0
	SM 1223 с 16 x 24 В DC вход / 16 x 24 В DC выход (потребитель)	6ES7223-1BL32-1XB0
	SM 1223 8 x вход 24 В DC (потребитель/источник) / 8 x релейный выход	6ES7223-1PH32-0XB0
	SM 1223 16 x вход 24 В DC (потребитель/источник) / 16 x релейный выход	6ES7223-1PL32-0XB0
	SM 1223 8 x вход 120/230 В AC (потребитель/источник) / 8 x релейный выход	6ES7223-1QH32-0XB0
Аналоговый ввод	SM 1231 4 x аналоговый вход	6ES7231-4HD32-0XB0
	SM 1231 8 x аналоговый вход	6ES7231-4HF32-0XB0
	SM 1231 4 x аналоговый вход x 16 бит (high feature)	6ES7231-5ND32-0XB0
	SM 1238 Energy Meter 480 В AC	6ES7238-5XA32-0XB0
Аналоговый вывод	SM 1232 2 x аналоговый выход	6ES7232-4HB32-0XB0
	SM 1232 4 x аналоговый выход	6ES7232-4HD32-0XB0
Аналоговый ввод/вывод	SM 1234 4 x аналоговый вход / 2 x аналоговый выход	6ES7234-4HE32-0XB0
RTD и TC	SM 1231 TC 4 x 16 бит	6ES7231-5QD32-0XB0
	SM 1231 TC 8 x 16 бит	6ES7231-5QF32-0XB0
	SM 1231 RTD 4 x 16 бит	6ES7231-5PD32-0XB0
	SM 1231 RTD 8 x 16 бит	6ES7231-5PF32-0XB0
Технологические модули	SM 1278 4xIO-Link-Master	6ES7278-4BD32-0XB0
	Модуль взвешивания SIWAREX WP231 с калибровкой (1 канал) для тензометрических весовых ячеек (1-4 MB/B) для SIMATIC S7-1200, интерфейс RS-485 и Ethernet, встроенные I/O: 4 DI / 4 DQ, 1 AQ (0/4...20 MA)	7MH4960-2AA01
	Модуль взвешивания SIWAREX WP241 для использования с SIMATIC S7-1200 на движущихся ленточных конвейерах (1 канал) для тензометрических весовых ячеек (1-4 MB/B), интерфейс RS-485 и Ethernet, встроенные I/O: 4 DI / 4 DQ, 1 AQ (0/4...20 MA)	7MH4960-4AA01
	Модуль взвешивания SIWAREX WP251 для автоматизации процессов дозирования и наполнения (1 канал) для тензометрических весовых ячеек (1-4 MB/B) с SIMATIC S7-1200, интерфейс RS-485 и Ethernet, встроенные I/O: 4 DI / 4 DQ, 1 AQ (0/4...20 MA)	7MH4960-6AA01

Таблица C-3 Сигнальные платы (SB) и батарейные платы (BB)

Сигнальные платы и батарейные платы		Заказной номер
Цифровой ввод	SB 1221 200 кГц, 4 x вход 24 В DC (источник)	6ES7221-3BD30-0XB0
	SB 1221 200 кГц, 4 x вход 5 В DC (источник)	6ES7221-3AD30-0XB0
Цифровой вывод	SB 1222 200 кГц, 4 x выход 24 В DC (потребитель/источник)	6ES7222-1BD30-0XB0
	SB 1222 200 кГц, 4 x выход 5 В DC (потребитель/источник)	6ES7222-1AD30-0XB0

Сигнальные платы и батарейные платы		Заказной номер
Цифровой ввод/вывод	SB 1223 с 2 x 24 В DC вход (потребитель) / 2 x 24 В DC выход (источник)	6ES7223-0BD30-0XB0
	SB 1223 200 кГц, 2 x вход 24 В DC (источник) / 2 x выход 24 В DC (потребитель/источник)	6ES7223-3BD30-0XB0
	SB 1223 200 кГц, 2 x вход 5 В DC (источник) / 2 x выход 5 В DC (потребитель/источник)	6ES7223-3AD30-0XB0
Аналоговые	SB 1232 1 аналоговый выход	6ES7232-4HA30-0XB0
	SB 1231 1 аналоговый вход	6ES7231-4HA30-0XB0
	SB 1231 с 1 аналоговым входом для термопары	6ES7231-5QA30-0XB0
	SB 1231 1 аналоговый вход RTD	6ES7231-5PA30-0XB0
Батарейная	BB 1297 батарейная плата (без батареи, тип CR1025)	6ES7297-0AX30-0XA0

C.3 Коммуникация

Таблица C-4 Коммуникационный модуль (CM)

Коммуникационный модуль (CM)			Заказной номер
RS232, RS422 и RS485	CM 1241 RS232	RS232	6ES7241-1AH32-0XB0
	CM 1241 RS422/485	RS422/485	6ES7241-1CH32-0XB0
PROFIBUS	CM 1243-5	Ведущее устройство PROFIBUS	6GK7243-5DX30-0XE0
	CM 1242-5	Ведомое устройство PROFIBUS	6GK7242-5DX30-0XE0
Ведущее устройство AS-i шины	CM 1243-2	Ведущее устройство AS-i шины	3RK7243-2AA30-0XB0

Таблица C-5 Коммуникационная плата (CB)

Коммуникационная плата (CB)			Заказной номер
RS485	CB 1241 RS485	RS485	6ES7241-1CH30-1XB0

Таблица C-6 Коммуникационный процессор (CP)

CP	Интерфейс	Заказной номер
CP 1242-7 GPRS V2	GPRS	6GK7242-7KX31-0XE0
CP 1243-7 LTE-US	LTE	6GK7243-7SX30-0XE0
CP 1243-7 LTE-EU	LTE	6GK7243-7KX30-0XE0
CP 1243-1	IE-интерфейс	6GK7243-1BX30-0XE0
CP 1243-8 IRC	IE- и последовательный интерфейс	6GK7243-8RX30-0XE0

Таблица C-7 Телесервис

TS адаптер	Заказной номер
TS адаптер IE Basic	6ES7972-0EB00-0XA0
TS адаптер IE Advanced	6ES7972-0EA00-0XA0
TS модуль GSM	6GK7972-0MG00-0XA0
TS модуль RS232	6ES7972-0MS00-0XA0
TS модуль модема	6ES7972-0MM00-0XA0
TS модуль ISDN	6ES7972-0MD00-0XA0

Таблица C-8 Аксессуары

Аксессуары			Заказной номер
Антенна	ANT794-4MR	GSM/GPRS антенна	6NH9860-1AA00
	ANT794-3M	Плоская антенна	6NH9870-1AA00

Таблица C-9 Соединители

Тип соединителя		Заказной номер
RS485	Винтовое клеммное соединение, отвод кабеля под углом 35 градусов	6ES7972-0BA42-0XA0
	Соединение FastConnect, отвод кабеля под углом 35 градусов	6ES7972-0BA60-0XA0

C.4 Оказобезопасные CPU и сигнальные модули

Таблица C-10 F-CPU

Модель F-CPU		Заказной номер
CPU 1212FC	CPU 1212FC DC/DC/DC	6ES7212-1AF40-0XB0
	CPU 1212FC DC/DC/RLY	6ES7212-1HF40-0XB0
CPU 1214FC	CPU 1214FC DC/DC/DC	6ES7214-1AF40-0XB0
	CPU 1214FC DC/DC/RLY	6ES7214-1HF40-0XB0
CPU 1215FC	CPU 1215FC DC/DC/DC	6ES7215-1AF40-0XB0
	CPU 1215FC DC/DC/RLY	6ES7215-1HF40-0XB0

Таблица C-11 Отказобезопасные сигнальные модули

Сигнальные модули для систем обеспечения безопасности		Заказной номер
Цифровой ввод	SM 1226 F-DI 16 x 24 V DC	6ES7226-6BA32-0XB0
Цифровой вывод	SM 1226 F-DQ 4 x 24 V DC	6ES7226-6DA32-0XB0
	SM 1226 F-DQ 2 x Relay	6ES7226-6RA32-0XB0

C.5 Прочие модули

Таблица C-12 Сопутствующие продукты

Описание		Заказной номер
Электропитание	Блок питания PM 1207	6EP1332-1SH71
Ethernet коммутатор	CSM 1277 Ethernet коммутатор - 4 порта	6GK7277-1AA10-0AA0
CM CANopen	CANopen для SIMATIC S7-1200	021620-B
	CANopen (Ruggedized) для SIMATIC S7-1200	021730-B
RF120C	Коммуникационный модуль RF120C	6GT2002-0LA00

C.6 Карты памяти

Таблица C-13 Карты памяти

Карты памяти SIMATIC	Заказной номер
SIMATIC MC 32 ГБ	6ES7954-8LT03-0AA0
SIMATIC MC 2 ГБ	6ES7954-8LP02-0AA0
SIMATIC MC 256 МБ	6ES7954-8LL03-0AA0
SIMATIC MC 24 МБ	6ES7954-8LF03-0AA0
SIMATIC MC 12 МБ	6ES7954-8LE03-0AA0
SIMATIC MC 4 МБ	6ES7954-8LC03-0AA0

C.7 Базовые устройства HMI

Таблица C-14 Устройства HMI

Базовые панели HMI	Заказной номер
KTP400 Basic (монохромная, PN)	6AV2123-2DB03-0AX0
KTP700 Basic	6AV2123-2GB03-0AX0
KTP700 Basic DP	6AV2123-2GA03-0AX0
KTP900 Basic	6AV2123-2JB03-0AX0
KTP1200 Basic	6AV2123-2MB03-0AX0
KTP1200 Basic DP	6AV2123-2MA03-0AX0

C.8 Запасные части и другое оборудование

Таблица C-15 Расширительные кабели, имитаторы и концевые фиксаторы

Описание		Заказной номер
Кабель для модулей расширения	Кабель для модулей расширения, 2 м	6ES7290-6AA30-0XA0
Имитатор ввода-вывода	Имитатор (1211C/1212C - 8 I/O)	6ES7274-1XF30-0XA0
	Имитатор (1214C/1215C - 14 I/O)	6ES7274-1XH30-0XA0
	Имитатор CPU 1217C	6ES7274-1XK30-0XA0
Модуль потенциометров	Модуль потенциометров S7-1200	6ES7274-1XA30-0XA0
Компенсатор натяжения Ethernet	Компенсатор натяжения для одиночного порта RJ45, 10/100 Мбит/с	6ES7290-3AA30-0XA0
	Компенсатор натяжения для двойного порта RJ45, 10/100 Мбит/с	6ES7290-3AB30-0XA0
Комплект защитной дверцы	CPU 1211C/1212C	6ES7291-1AA30-0XA0
	CPU 1214C	6ES7291-1AB30-0XA0
	CPU 1215C	6ES7291-1AC30-0XA0
	CPU 1217C	6ES7291-1AD30-0XA0
	Сигнальный модуль (SM), 45 мм	6ES7291-1BA30-0XA0
	Сигнальный модуль (SM), 70 мм	6ES7291-1BB30-0XA0
	Коммуникационный модуль (для использования с модулями 6ES72xx-xxx32-0XB0 и 6ES72xx-xxx30-0XB0)	6ES7291-1CC30-0XA0
Концевой фиксатор	Термопластиковый концевой фиксатор, 10 мм	8WA1808
	Стальной концевой фиксатор, 10,3 мм	8WA1805

Замена терминального блока

Важно, чтобы для модуля использовался правильный терминальный блок. Для выбора правильного терминального блока следует использовать таблицы ниже и технические данные используемого модуля.

Примечание

Съемные терминальные блоки с устройством механического кодирования

Правильное подключение PLC является важным условием для безопасной и надежной работы.

При замене терминального блока в CPU или на SM важно, чтобы были выбраны правильный терминальный блок и правильный источник подключения.

Кодирование помогает предотвратить случайную установку подключенного к высокому напряжению терминального блока в низковольтный модуль или установку подключенного к особому напряжению терминального блока в модуль с обычным напряжением. Устройство механического кодирования располагается слева, справа или в центре терминальных блоков.

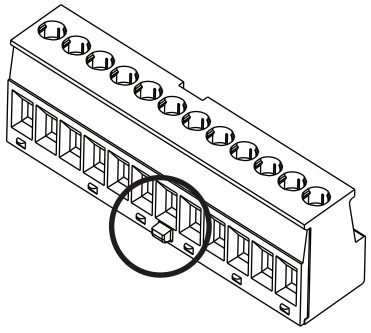
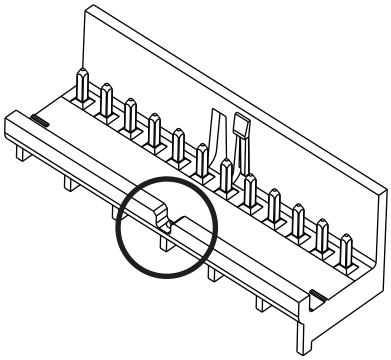
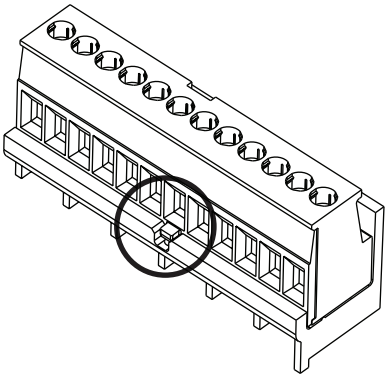
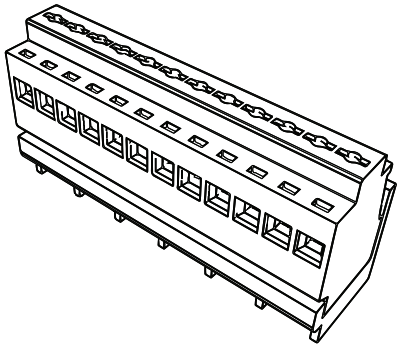
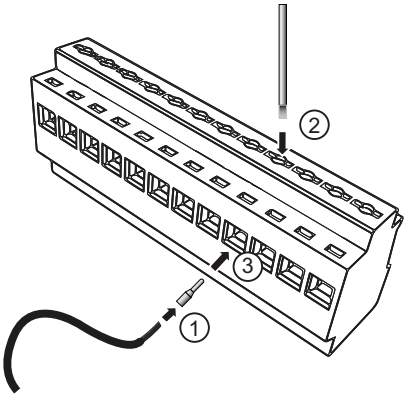
<table>
<tr><td></td><td></td><td></td></tr>
<tr><td>Съемный терминальный блок (пример с механическим кодированием)</td><td>Кодирование на устройстве</td><td>Кодирование на устройстве точно соответствует определенному терминальному блоку</td></tr>
<tr><td></td><td colspan="2">
</td></tr>
<tr><td>Push-in терминальный блок</td><td colspan="2">Выполнить следующие действия:
1. Вставить провод и обжимной контакт.
2. С помощью маленькой отвертки нажать на язычок, чтобы открыть слот.
3. Вставить обжимное соединение в открытый слот на терминальном блоке.</td></tr>
</table>

Таблица C-16 S7-1200 CPU от версии V4 - Запасные терминальные блоки

Если используется S7-1200 CPU от версии V4.0 (заказной номер)	**В этом случае применяются следующие запасные терминальные блоки (4/упаковка)**		
	Терминальный блок (клеммы под винт) Заказной номер	**Эквивалент Push-in терминальный блок Заказной номер**	**Терминальный блок Описание**
CPU 1211C DC/DC/DC (6ES7211-1AE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-полюсный терминальный блок с позолоченными контактами
	6ES7292-1AH30-0XA0	6ES7292-2AH30-0XA0	8-полюсный терминальный блок с луженными контактами
	6ES7292-1AP30-0XA0	6ES7292-2AP30-0XA0	14-полюсный терминальный блок с луженными контактами
CPU 1211C DC/DC/RLY (6ES7211-1HE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-полюсный терминальный блок с позолоченными контактами
	6ES7292-1AH40-0XA0	6ES7292-2AH40-0XA0	8-полюсный терминальный блок с луженными контактами
	6ES7292-1AP30-0XA0	6ES7292-2AP30-0XA0	14-полюсный терминальный блок с луженными контактами
CPU 1211C AC/DC/RLY (6ES7211-1BE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-полюсный терминальный блок с позолоченными контактами
	6ES7292-1AH40-0XA0	6ES7292-2AH40-0XA0	8-полюсный терминальный блок с луженными контактами
	6ES7292-1AP40-0XA0	6ES7292-2AP40-0XA0	14-полюсный терминальный блок с луженными контактами, с устройством механического кодирования
CPU 1212C DC/DC/DC (6ES7212-1AE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-полюсный терминальный блок с позолоченными контактами
	6ES7292-1AH30-0XA0	6ES7292-2AH30-0XA0	8-полюсный терминальный блок с луженными контактами
	6ES7292-1AP30-0XA0	6ES7292-2AP30-0XA0	14-полюсный терминальный блок с луженными контактами

Если используется S7-1200 CPU от версии V4.0 (заказной номер)	**В этом случае применяются следующие запасные терминальные блоки (4/упаковка)**		
	Терминальный блок (клеммы под винт) Заказной номер	**Эквивалент Push-in терминальный блок Заказной номер**	**Терминальный блок Описание**
CPU 1212C DC/DC/RLY (6ES7212-1HE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-полюсный терминальный блок с позолоченными контактами
	6ES7292-1AH40-0XA0	6ES7292-2AH40-0XA0	8-полюсный терминальный блок с луженными контактами
	6ES7292-1AP30-0XA0	6ES7292-2AP30-0XA0	14-полюсный терминальный блок с луженными контактами
CPU 1212C AC/DC/RLY (6ES7212-1BE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-полюсный терминальный блок с позолоченными контактами
	6ES7292-1AH40-0XA0	6ES7292-2AH40-0XA0	8-полюсный терминальный блок с луженными контактами
	6ES7292-1AP40-0XA0	6ES7292-2AP40-0XA0	14-полюсный терминальный блок с луженными контактами, с устройством механического кодирования
CPU 1214C DC/DC/DC (6ES7214-1AG40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-полюсный терминальный блок с позолоченными контактами
	6ES7292-1AM30-0XA0	6ES7292-2AM30-0XA0	12-полюсный терминальный блок с луженными контактами
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20-полюсный терминальный блок с луженными контактами
CPU 1214C DC/DC/RLY (6ES7214-1HG40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-полюсный терминальный блок с позолоченными контактами
	6ES7292-1AM40-0XA0	6ES7292-2AM40-0XA0	12-полюсный терминальный блок с луженными контактами, с устройством механического кодирования
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20-полюсный терминальный блок с луженными контактами

Если используется S7-1200 CPU от версии V4.0 (заказной номер)	**В этом случае применяются следующие запасные терминальные блоки (4/упаковка)**		
	Терминальный блок (клеммы под винт) Заказной номер	**Эквивалент Push-in терминальный блок Заказной номер**	**Терминальный блок Описание**
CPU 1214C AC/DC/RLY (6ES7214-1BG40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-полюсный терминальный блок с позолоченными контактами
	6ES7292-1AM40-0XA0	6ES7292-2AM40-0XA0	12-полюсный терминальный блок с луженными контактами, с устройством механического кодирования
	6ES7292-1AV40-0XA0	6ES7292-2AV40-0XA0	20-полюсный терминальный блок с луженными контактами, с устройством механического кодирования
CPU 1215C DC/DC/DC (6ES7215-1AG40-0XB0)	6ES7292-1BF30-0XB0	6ES7292-2BF30-0XB0	6-полюсный терминальный блок с позолоченными контактами
	6ES7292-1AM30-0XA0	6ES7292-2AM30-0XA0	12-полюсный терминальный блок с луженными контактами
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20-полюсный терминальный блок с луженными контактами
CPU 1215C DC/DC/RLY (6ES7215-1HG40-0XB0)	6ES7292-1BF30-0XB0	6ES7292-2BF30-0XB0	6-полюсный терминальный блок с позолоченными контактами
	6ES7292-1AM40-0XA0	6ES7292-2AM40-0XA0	12-полюсный терминальный блок с луженными контактами, с устройством механического кодирования
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20-полюсный терминальный блок с луженными контактами

Если используется S7-1200 CPU от версии V4.0 (заказной номер)	**В этом случае применяются следующие запасные терминальные блоки (4/упаковка)**		
	Терминальный блок (клеммы под винт) Заказной номер	**Эквивалент Push-in терминальный блок Заказной номер**	**Терминальный блок Описание**
CPU 1215C AC/DC/RLY (6ES7215-1BG40-0XB0)	6ES7292-1BF30-0XB0	6ES7292-2BF30-0XB0	6-полюсный терминальный блок с позолоченными контактами
	6ES7292-1AM40-0XA0	6ES7292-2AM40-0XA0	12-полюсный терминальный блок с луженными контактами, с устройством механического кодирования
	6ES7292-1AV40-0XA0	6ES7292-2AV40-0XA0	20-полюсный терминальный блок с луженными контактами, с устройством механического кодирования
CPU 1217C DC/DC/DC (6ES7217-1AG40-0XB0)	6ES7292-1BF30-0XB0	6ES7292-2BF30-0XB0	6-полюсный терминальный блок с позолоченными контактами
	6ES7292-1AK30-0XA0	6ES7292-2AK30-0XA0	10-полюсный терминальный блок с луженными контактами
	6ES7292-1AR30-0XA0	6ES7292-2AR30-0XA0	16-полюсный терминальный блок с луженными контактами
	6ES7292-1AT30-0XA0	6ES7292-2AT30-0XA0	18-полюсный терминальный блок с луженными контактами

Таблица C-17 S7-1200 SM от версии V3.2 - Запасные терминальные блоки

Если используются S7-1200 SM от версии V3.2 (заказной номер)	**В этом случае применяются следующие запасные терминальные блоки (4/упаковка)**		
	Терминальный блок (с винтовыми клеммами) Заказной номер	**Эквивалент Push-in терминальный блок Заказной номер**	**Терминальный блок Описание**
SM 1221 DI 8 x DC (6ES7221-1BF32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7-полюсный терминальный блок с луженными контактами
SM 1222 DQ 8 x DC (6ES7222-1BF32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7-полюсный терминальный блок с луженными контактами

Если используются S7-1200 SM от версии V3.2 (заказной номер)	**В этом случае применяются следующие запасные терминальные блоки (4/упаковка)**		
	Терминальный блок (с винтовыми клеммами) Заказной номер	**Эквивалент Push-in терминальный блок Заказной номер**	**Терминальный блок Описание**
SM 1222 DQ 8 x RLY (6ES7222-1HF32-0XB0)	6ES7292-1AG40-0XA1	6ES7292-2AG40-0XA1	7-полюсный терминальный блок с луженными контактами, с устройством механического кодирования с левой стороны
SM 1238 Energy Meter 480 В AC (6ES7238-5XA32-0XB0) для входа по напряжению (сверху)	6ES7292-1AG40-0XA2	6ES7292-2AG40-0XA2	7-полюсный терминальный блок с луженными контактами, с устройством механического кодирования по центру
SM 1238 Energy Meter 480 В AC (6ES7238-5XA32-0XB0) для входа по току (снизу)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7-полюсный терминальный блок с луженными контактами
SM 1231 AI 4 x 13 бит (6ES7231-4HD32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-полюсный терминальный блок с позолоченными контактами
SM 1232 AQ 2 x 14 бит (6ES7232-4HB32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-полюсный терминальный блок с позолоченными контактами
SM 1231 AI 4 x TC (6ES7231-5QD32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-полюсный терминальный блок с позолоченными контактами
SM 1231 AI 4 x 16 бит (6ES7231-5ND32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-полюсный терминальный блок с позолоченными контактами
SM 1221 DI 16 x DC (6ES7221-1BH32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7-полюсный терминальный блок с луженными контактами
SM 1222 DQ 16 x DC (6ES7222-1BH32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7-полюсный терминальный блок с луженными контактами
SM 1222 DQ 16 x RLY (6ES7222-1HH32-0XB0)	6ES7292-1AG40-0XA0	6ES7292-2AG40-0XA0	7-полюсный терминальный блок с луженными контактами, с устройством механического кодирования с правой стороны

Если используются S7-1200 SM от версии V3.2 (заказной номер)	**В этом случае применяются следующие запасные терминальные блоки (4/упаковка)**		
	Терминальный блок (с винтовыми клеммами) Заказной номер	**Эквивалент Push-in терминальный блок Заказной номер**	**Терминальный блок Описание**
SM 1223 DI 8 x DC/DQ 8 x DC (6ES7223-1BH32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7-полюсный терминальный блок с луженными контактами
SM 1223 8 x DC/8 x RLY (6ES7223-1PH32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7-полюсный терминальный блок с луженными контактами
	6ES7292-1AG40-0XA0	6ES7292-2AG40-0XA0	7-полюсный терминальный блок с луженными контактами, с устройством механического кодирования с правой стороны
SM 1223 8 x AC/8 x RLY (6ES7223-1QH32-0XB0)	6ES7292-1AG40-0XA0	6ES7292-2AG40-0XA0	7-полюсный терминальный блок с луженными контактами, с устройством механического кодирования с правой стороны
SM 1234 AI 4 / AQ 2 (6ES7234-4HE32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-полюсный терминальный блок с позолоченными контактами
SM 1231 AI 8 x 13 бит (6ES7231-4HF32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-полюсный терминальный блок с позолоченными контактами
SM 1232 AQ 4 x 14 бит (6ES7232-4HD32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-полюсный терминальный блок с позолоченными контактами
SM 1231 AI 4 x RTD (6ES7231-5PD32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-полюсный терминальный блок с позолоченными контактами
SM 1231 AI 8 x TC (6ES7231-5QF32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-полюсный терминальный блок с позолоченными контактами
SM 1278 IO LINK (6ES7278-4BD32 0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7-полюсный терминальный блок с луженными контактами

Если используются S7-1200 SM от версии V3.2 (заказной номер)	**В этом случае применяются следующие запасные терминальные блоки (4/упаковка)**		
	Терминальный блок (с винтовыми клеммами) Заказной номер	**Эквивалент Push-in терминальный блок Заказной номер**	**Терминальный блок Описание**
SM 1222 DQ 8 x RLY (переключение) (6ES7222-1XF32-0XB0)	6ES7292-1AL30-0XA0	6ES7292-2AL30-0XA0	11-полюсный терминальный блок с луженными контактами
SM 1223 DI 16 x DC/DQ 16 x DC (6ES7223-1BL32-0XB0)	6ES7292-1AL30-0XA0	6ES7292-2AL30-0XA0	11-полюсный терминальный блок с луженными контактами
SM 1223 DI 16 x DC/DQ 16 x RLY (6ES7223-1PL32-0XB0)	6ES7292-1AL30-0XA0	6ES7292-2AL30-0XA0	11-полюсный терминальный блок с луженными контактами
	6ES7292-1AL40-0XA0	6ES7292-2AL40-0XA0	11-полюсный терминальный блок с луженными контактами, с устройством механического кодирования
SM 1231 AI 8 x RTD (6ES7231-5PF32-0XB0)	6ES7292-1BL30-0XA0	6ES7292-2BL30-0XA0	11-полюсный терминальный блок с позолоченными контактами

Таблица C-18 S7-1200 SB, CB и BB - Запасные терминальные блоки

Если используется S7-1200 SB, CB или BB (заказной номер)	**В этом случае применяются следующие запасные терминальные блоки (4/упаковка)**	
	Терминальный блок (с винтовыми клеммами) Заказной номер	**Терминальный блок Описание**
SB 1221 DI 4 x 5 VDC (6ES7221-3AD30-0XB0)	6ES7292-1BF30-0XA0	6-полюсный
SB 1221 DI 4 x 5 VDC (6ES7221-3AD30-0XB0)		
SB 1221 DI 4 x 24 VDC (6ES7221-3BD30-0XB0)		
SB 1222 DO 4 x 5 VDC (6ES7222-1AD30-0XB0)		
SB 1222 DO 4 x 24 VDC (6ES7222-1BD30-0XB0)		
SB 1223 DI 2 x 24 VDC / DQ 2 x 24 VDC (6ES7223-0BD30-0XB0)		
SB 1223 DI 2 x 5 VDC / DQ 2 x 5 VDC (6ES7223-3AD30-0XB0)		
SB 1223 DI 2 x 24 VDC / DQ 2 x 24 VDC (6ES7223-3BD30-0XB0)		
SB 1231 AI 1 x 12 бит (6ES7231-4HA30-0XB0)		
SB 1231 AI 1 x RTD (6ES7231-5PA30-0XB0)		
SB 1231 AI 1 x TC (6ES7231-5QA30-0XB0)		
SB 1232 AQ 1 x 12 бит (6ES7232-4HA30-0XB0)		
CB 1231 RS-485 (6ES7241-1CH30-1XB0)		
BB 1297 батарея (6ES7297-0AX30-0XA0)		

Таблица C-19 Отказобезопасные CPU - Запасные терминальные блоки

Если используется отказобезопасный CPU (заказной номер)	**В этом случае применяются следующие запасные терминальные блоки (4/упаковка)**	
	Терминальный блок Заказной номер	**Терминальный блок Описание**
CPU 1214FC DC/DC/DC (6ES7214-1AF40-0XB0)	6ES7292-1BC30-0XA0	3-полюсный терминальный блок с позолоченными контактами
	6ES7292-1AM30-0XA0	12-полюсный терминальный блок с лужеными контактами
	6ES7292-1AV30-0XA0	20-полюсный терминальный блок с лужеными контактами

Если используется отказобезопасный CPU (заказной номер)	**В этом случае применяются следующие запасные терминальные блоки (4/упаковка)**	
	Терминальный блок Заказной номер	**Терминальный блок Описание**
CPU 1214FC DC/DC/RLY (6ES7214-1HF40-0XB0)	6ES7292-1BC30-0XA0	3-полюсный терминальный блок с позолоченными контактами
	6ES7292-1AM40-0XA0	12-полюсный терминальный блок с луженными контактами, с устройством механического кодирования
	6ES7292-1AV30-0XA0	20-полюсный терминальный блок с луженными контактами
CPU 1215FC DC/DC/DC (6ES7215-1AF40 0XB0)	6ES7292-1BF30-0XB0	6-полюсный терминальный блок с позолоченными контактами
	6ES7292-1AM30-0XA0	12-полюсный терминальный блок с луженными контактами
	6ES7292-1AV30-0XA0	20-полюсный терминальный блок с луженными контактами
CPU 1215FC DC/DC/RLY (6ES7215-1HF40 0XB0)	6ES7292-1BF30-0XB0	6-полюсный терминальный блок с позолоченными контактами
	6ES7292-1AM40-0XA0	2-полюсный терминальный блок с луженными контактами, с устройством механического кодирования
	6ES7292-1AV30-0XA0	20-полюсный терминальный блок с луженными контактами

Таблица C-20 Отказобезопасные сигнальные модули - Запасные терминальные блоки

Если используется отказобезопасный сигнальный модуль (заказной номер)	**В этом случае применяются следующие запасные терминальные блоки (4/упаковка)**	
	Терминальный блок Заказной номер	**Терминальный блок Описание**
SM 1226 F-DI (6ES7226-6BA32-0XB0)	6ES7292-1AL30-0XA0	11-полюсный терминальный блок с луженными контактами
SM 1226 F-DQ (6ES7226-6DA32-0XB0)	6ES7292-1AL30-0XA0	11-полюсный терминальный блок с луженными контактами
SM 1226 F-RLY (6ES7226-6RA32-0XB0)	6ES7292-1AL40-0XA0	11-полюсный терминальный блок с луженными контактами, с устройством механического кодирования

C.9 ПО для программирования

Таблица C-21 ПО для программирования

Программное обеспечение SIMATIC		**Заказной номер**
ПО для программирования	STEP 7 Basic V16	6ES7822-0AA06-0YA5
	STEP 7 Professional V16	6ES7822-1AA06-0YA5
Программное обеспечение визуализации	WinCC Basic V16	6AV2100-0AA06-0AA5
	WinCC Comfort V16	6AV2101-0AA06-0AA5
	WinCC Advanced V16	6AV2102-0AA06-0AA5
	WinCC Professional 512 PowerTags V16	6AV2103-0DA06-0AA5
	WinCC Professional 4096 PowerTags V16	6AV2103-0HA06-0AA5
	WinCC Professional max. PowerTags V16	6AV2103-0XA06-0AA5

C.10 Лицензии OPC UA

Таблица C-22 Лицензии OPC UA для S7-1200

Лицензии OPC UA		**Заказной номер**
Лицензии SIMATIC	SIMATIC OPC UA S7-1200 Basic DVD	6ES7823-0BA00-2BA0
	SIMATIC OPC UA S7-1200 Basic DL	6ES7823-0BE00-2BA0

Замена устройств и совместимость по запасным частям

D

D.1 Замена CPU V3.0 на CPU V4.x

Для модернизации CPU версии V3.0 до CPU версии V4.x необходимо заменить аппаратное обеспечение CPU. Обновления прошивки недостаточно, чтобы выполнить апгрейд CPU V3.0 до CPU V4.x.

После в проекте STEP 7 можно заменить используемый CPU V3.0 на CPU V4.x (Страница 166) и использовать свой существующий, разработанный для CPU V3.0 проект STEP 7.

При замене CPU V3.0 на CPU V4.x, можно одновременно выполнить поиск обновления прошивки (Страница 147) для сигнальных и коммуникационных модулей и установить его.

Примечание

Замена устройства в STEP 7 из версии V4.x на V3.0 невозможна

Можно заменить CPU V3.0 на CPU V4.x, но после загрузки конфигурации нельзя заменить CPU V4.x на CPU V3.0. Для отображения или иного использования существующего проекта STEP 7 V3.0, перед заменой устройства необходимо создать архив проекта STEP 7 V3.0.

Если конфигурация с заменой устройства еще не была загружена, то возможна отмена. После загрузки уже нельзя отменить замену V3.0 на V4.x.

Необходимо учитывать различия в конфигурации и эксплуатационных характеристиках между этими версиями CPU:

Обновление проекта STEP 7

Нельзя сразу обновить проекты STEP 7 версии V11 или V12 до версии STEP 7 V15. Сначала следует обновить эти проекты до STEP 7 V13 SP1 или STEP 7 V13 SP2. После можно использовать такой проект в качестве базы для апгрейда до STEP 7 V15.

⚠ ПРЕДУПРЕЖДЕНИЕ

Риски при копировании и вставке логики программы из старых версий STEP 7

Копирование логики программы из старой версии STEP 7, напр., STEP 7 V12 в STEP 7 V15, может стать причиной непредсказуемого поведения при выполнении программы или ошибок при компиляции. В разных версиях STEP 7 элементы программы реализуются по разному. При компиляции различия обнаруживаются не всегда, если изменения были внесены путем вставки из старой версии в STEP 7 V15. Выполнение непредсказуемой логики программы может стать причиной тяжких или опасных для жизни телесных повреждений, если программа не будет исправлена.

При использовании логики программы из версии STEP 7 до версии STEP 7 V15 всегда следует обновлять весь проект до версии STEP 7 V15. При необходимости можно копировать, вырезать, вставлять и редактировать логику программы. В STEP 7 V15 можно открыть проект из STEP 7 V13 SP1 или выше. После STEP 7 выполняет необходимые для обеспечения совместимости преобразования и правильно обновляет программу. Такие преобразования и исправления в комбинации с обновлением необходимы для правильной компиляции и выполнения программы. Если проект старше, чем STEP 7 V13 SP1, то следует пошагово обновить проект до STEP 7 V15.

Организационные блоки

Можно настроить выполнение OB как прерываемое или непрерываемое (Страница 103). В проектах CPU версии V3.0 программа STEP 7 по умолчанию устанавливает для всех OB непрерываемое выполнение.

STEP 7 устанавливает для всех приоритетов OB (Страница 103) значения из STEP 7 проекта V3.0-CPU.

В дальнейшем при необходимости можно изменить настройки для прерываемости и приоритетов.

Стартовая информация OB диагностического прерывания (Страница 96) относится к субмодулю в целом, если нет активного диагностического события

Защита CPU паролем

STEP 7 устанавливает для CPU V4.x тот жеуровень парольной защиты (Страница 203), который был установлен в CPU V3.0, и назначает пароль от версии 3.0 в качестве пароля для "Полного доступа (защита отсутствует)" для CPU V4.x:

Уровень защиты в версии 3.0	Уровень защиты в версии 4.x
Защита отсутствует	Полный доступ (защита отсутствует)
Защита от записи	Доступ по чтению
Защита от записи/чтения	Доступ через HMI

Необходимо учитывать, что уровня доступа в V4.x типа "Доступ отсутствует (полная защита)" нет в версии 3.0.

Веб-сервер

При использовании в проекте V3.0 пользовательских веб-страниц, следует сохранить их в папке установки проекта в подпапке "UserFiles\Webserver" перед запуском апгрейда проекта. Если пользовательские страницы находятся в этой папке, то при сохранении проекта STEP 7 будут сохранены и пользовательские страницы.

При замене CPU V3.0 на CPU V4.x, настройки проекта веб-сервера (Страница 1026) для активации веб-сервера и установок HTTPS совпадают с таковыми в версии 3.0. После можно изменить пользователей, права, пароли (Страница 1028) и языки (Страница 1026) для работы с веб-сервером. При конфигурировании пользователей без дополнительных прав, права просмотра ограничиваются информацией на стандартных веб-страницах (Страница 1036). S7-1200 CPU V4.x не поддерживает ранее сконфигурированного пользователя "admin" и соответствующий пароль.

Страница веб-сервера для архива данных S7-1200 V3.0 ранее предлагала функцию "Загрузка и очистка". Страница браузера файлов (Страница 1068) веб-сервера версии 4.x, через которую выполняется доступ к архивам данных, больше не предлагает эту функцию. Вместо этого веб-сервер позволяет загружать, переименовывать и удалять файлы архивов данных.

Несовместимость трансферных карт

Нельзя использовать трансферную карту (Страница 137) версии V3.0, чтобы перенести программу V3.0 в CPU V4.x. Необходимо открыть проект V3.0 в STEP 7, изменить устройство на CPU V4.x (Страница 166) и после загрузить проект STEP 7 в CPU V4.x. После изменения проекта на проект V4.x, можно сделать трансферную карту V4.x для последующих переносов программы.

Коммуникация GET/PUT

По умолчанию у S7-1200 V3.0-CPU коммуникация через GET/PUT активирована.
При замене CPU V3.0 на CPU V4.x (Страница 166), в области информации о совместимости появляется сообщение, что GET/PUT активированы.

Поддержка управления перемещением

S7-1200 CPU V4.x не поддерживают библиотеки управления перемещением V1.0 и V2.0. Если выполняется замена устройства для проекта STEP 7 с библиотеками управления перемещением V1.0 или V2.0, то в ходе этого процесса они заменяются инструкциями управления перемещением (Страница 735) V3.0, совместимыми на момент компиляции.

Если выполняется замена устройства CPU V3.0 на CPU V4.x для проекта STEP 7 с двумя различными версиями инструкций управления перемещением (V3.0 и V5.0), то в ходе этого процесса они заменяются инструкциями управления перемещением (Страница 735) версии 5.0, совместимыми на момент компиляции.

При замене устройства CPU V3.0 на CPU V4.x, версия технологического объекта (ТО) для управления перемещением не изменяется автоматически с V3.0 на V5.0. Если

необходимо выполнить обновление до более поздних версий, то следует перейти к дереву инструкций и выбрать требуемую версию управления перемещением S7-1200 для проекта. Это показано в таблице ниже:

Версия CPU	Допустимые версии управления перемещением
V4.3 (управление перемещением V5.0)	V6.0 или V5.0 или V4.0 или V3.0
V4.2.x (управление перемещением V5.0)	V6.0 или V5.0 или V4.0 или V3.0
V4.1 (управление перемещением V5.0)	V5.0 или V4.0 или V3.0
V4.0 (управление перемещением V4.0)	V4.0 или V3.0
V3.0 (управление перемещением V3.0)	V3.0

Существуют различия в структуре для ТО управления перемещением версий V3.0 и V5.0. Все связанные блоки также изменяются. Интерфейсы блоков, таблицы текущего контроля и трассировки обновляются согласно новой структуре V5.0 для управления перемещением. Отличия между параметрами управления перемещением для оси между CPU V3.0 и CPU V4.x представлены в следующих двух таблицах:

CPU V3.0 (управление перемещением V3.0)	CPU V4.x (управление перемещением V5.0)
Config.General.LengthUnit	Units.LengthUnit
Config.Mechanics.PulsesPerDriveRevolution	Actor.DriveParameter.PulsesPerDriveRevolution
Config.Mechanics.LeadScrew	Mechanics.LeadScrew
Config.Mechanics.InverseDirection	Actor.InverseDirection
Config.DynamicLimits.MinVelocity	DynamicLimits.MinVelocity
Config.DynamicLimits.MaxVelocity	DynamicLimits.MaxVelocity
Config.DynamicDefaults.Acceleration	DynamicDefaults.Acceleration
Config.DynamicDefaults.Deceleration	DynamicDefaults.Deceleration
Config.DynamicDefaults.EmergencyDeceleration	DynamicDefaults.EmergencyDeceleration
Config.DynamicDefaults.Jerk	DynamicDefaults.Jerk
Config.PositionLimits_SW.Active	PositionLimitsSW.Active
Config.PositionLimits_SW.MinPosition	PositionLimitsSW.MinPosition
Config.PositionLimits_SW.MaxPosition	PositionLimitsSW.MaxPosition
Config.PositionLimits_HW.Active	PositionLimitsHW.Active
Config.PositionLimits_HW.MinSwitchedLevel	PositionLimitsHW.MinSwitchLevel
Config.PositionLimits_HW.MaxSwitchedLevel	PositionLimitsHW.MaxSwitchLevel
Config.Homing.AutoReversal	Homing.AutoReversal
Config.Homing.Direction	Homing.ApproachDirection
Config.Homing.SideActiveHoming	Sensor[1].ActiveHoming.SideInput
Config.Homing.SidePassiveHoming	Sensor[1].PassiveHoming.SideInput
Config.Homing.Offset	Sensor[1].ActiveHoming.HomePositionOffset
Config.Homing.FastVelocity	Homing.ApproachVelocity
Config.Homing.SlowVelocity	Homing.ReferencingVelocity
MotionStatus.Position	Position
MotionStatus.Velocity	Velocity
MotionStatus.Distance	StatusPositioning.Distance

CPU V3.0 (управление перемещением V3.0)	CPU V4.x (управление перемещением V5.0)
MotionStatus.TargetPosition	StatusPositioning.TargetPosition
StatusBits.SpeedCommand	StatusBits.VelocityCommand
StatusBits.Homing	StatusBits.HomingCommand

Единственный параметр "commandtable", который переименован, является массивом с командами:

V3.0	V4.x
Config.Command[]	Command[]

Примечание: Массив "Command[]" представляет собой UDT с типом "TO_CmdTab_Config_Command" в V3.0 и "TO_Struct_Command" в V4.x.

Измененные инструкции

Следующие инструкции имеют изменения в параметрах или поведении:

- RDREC и WRREC (Страница 387)
- CONV (Страница 299)

Коммуникация с HMI панелью

Если одна или несколько HMI панелей (Страница 32) были подключены к S7-1200 CPU версии 3.0, то обмен данными с S7-1200 CPU версии 4.x зависит от используемого типа коммуникации и от версии прошивки панели HMI. Перекомпилировать проект и загрузить его в CPU и HMI и/или обновить прошивку HMI.

Необходимые условия для перекомпиляции программных блоков

После замены CPU V3.0 на CPU V4.x необходимо перекомпилировать все программные блоки, прежде чем можно будет загрузить их в CPU V4.x. Кроме этого, если у какого-либо из блоков есть защита ноу-хау (Страница 206) или защита от копирования, привязанная к серийному номеру PLC (Страница 207), необходимо снять защиту, прежде чем компилировать и загружать блоки. (Можно не деактивировать защиту от копирования, связанную с картой памяти.) После успешной компиляции можно снова сконфигурировать защиту ноу-хау и/или защиту от копирования с привязкой к серийному номеру PLC. Следует помнить, что если в проекте есть какие-либо блоки с защитой ноу-хау, предоставленные OEM (производитель оборудования), необходимо связаться с OEM относительно возможности получения версии V4.x для этих блоков.

Как правило, Siemens рекомендует выполнить перекомпилирование конфигурации оборудования и программного обеспечения в STEP 7 и их загрузку во все устройства в проекте после замены устройства. Исправить все ошибки, которые обнаруживаются при компиляции проекта, и повторить компиляцию до их полного устранения. После можно загрузить проект в CPU версии 4.x.

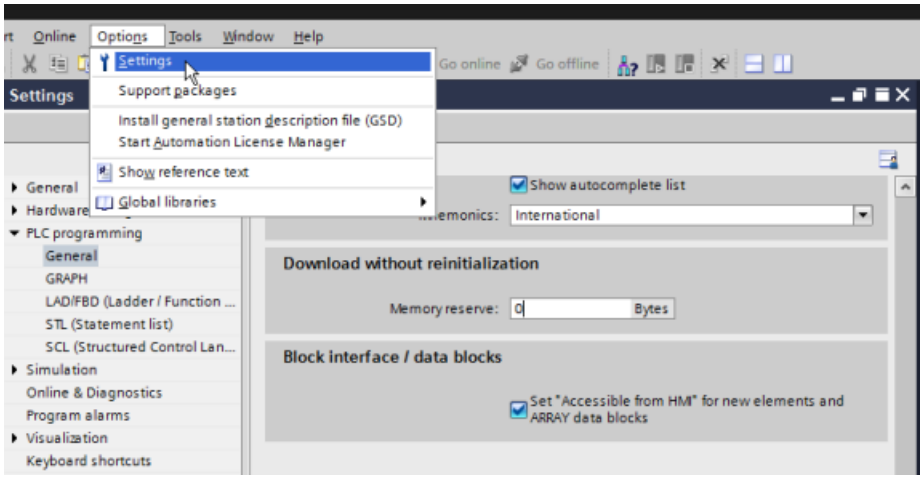
Проекты S7-1200 V3.0 не поддерживаются в S7-1200 CPU V4.x.

S7-1200 от V4.0 добавил резервную область в 100 байтов в каждый DB, чтобы поддержать загрузку без повторной инициализации.

Можно удалить 100-байтовую резервную область из DB до попытки загрузки проекта V3.0 в CPU V4.x.

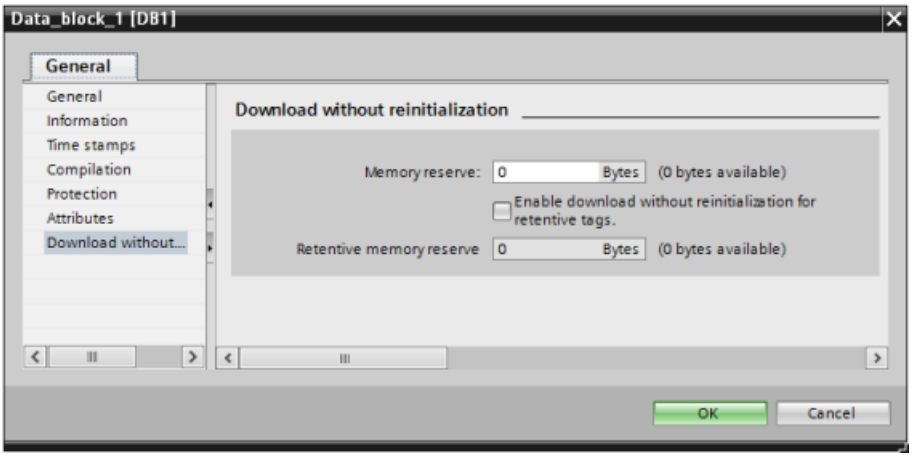
Чтобы удалить 100-байтовую резервную область, выполнить следующие шаги перед заменой устройства:

1. Выбрать в главном меню в TIA-Portal команду меню "Опции" > "Настройки".
2. Открыть в дереве проекта узел "Программирование PLC" > "Общее".
3. Задать в области "Загрузка без повторной инициализации" резерв памяти в 0 байт.



Если замена устройства уже выполнена, то придется удалять 100-байтовый резерв в каждом блоке по отдельности:

1. В папке программных блоков в дереве проекта кликнуть правой кнопкой по блоку и выбрать "Свойства" из контекстного меню.
2. Выбрать в диалоговом окне свойств блока данных узел "Загрузка без повторной инициализации".
3. Задать резерв памяти в 0 байт.
4. Повторить этот процесс для каждого блока данных в проекте.



D.2 S7-1200 до V3.0 - Запасные терминальные блоки

Таблица D-1 S7-1200 CPU до V3.0 - Запасные терминальные блоки

Имеется S7-1200 CPU до V3.0 (заказной номер)	**В этом случае применяются следующие запасные терминальные блоки (4/упаковка)**	
	Заказной номер терминального блока	**Описание терминального блока**
CPU 1211C DC/DC/DC (6ES7211-1AE31-0XB0) CPU 1211C AC/DC/RLY (6ES7211-1BE31-0XB0) CPU 1211C DC/DC/RLY (6ES7211-1HE31-0XB0) CPU 1212C DC/DC/DC (6ES7212-1AE31-0XB0) CPU 1212C AC/DC/RLY (6ES7212-1BE31-0XB0) CPU 1212C DC/DC/RLY (6ES7212-1HE31-0XB0)	6ES7292-1BC30-0XA0 6ES7292-1AH30-0XA0 6ES7292-1AP30-0XA0	3-полюсный терминальный блок с позолоченными контактами 8-полюсный терминальный блок с позолоченными контактами 14-полюсный терминальный блок с луженными контактами
CPU 1214C DC/DC/DC (6ES7214-1AG31-0XB0) CPU 1214C AC/DC/RLY (6ES7214-1BG31-0XB0) CPU 1214C DC/DC/RLY (6ES7214-1HG31-0XB0)	6ES7292-1BC30-0XA0 6ES7292-1AM30-0XA0 6ES7292-1AV30-0XA0	3-полюсный терминальный блок с позолоченными контактами 12-полюсный терминальный блок с луженными контактами 20-полюсный терминальный блок с луженными контактами

Имеется S7-1200 CPU до V3.0 (заказной номер)	**В этом случае применяются следующие запасные терминальные блоки (4/упаковка)**	
	Заказной номер терминального блока	**Описание терминального блока**
CPU 1215C DC/DC/DC (6ES7215-1AG31-0XB0)	6ES7292-1BF30-0XB0 6ES7292-1AM30-0XA0 6ES7292-1AV30-0XA0	6-полюсный терминальный блок с позолоченными контактами 12-полюсный терминальный блок с луженными контактами 20-полюсный терминальный блок с луженными контактами
CPU 1215C AC/DC/RLY (6ES7215-1BG31-0XB0)		
CPU 1215C DC/DC/RLY (6ES7215-1HG31-0XB0)		

Таблица D-2 S7-1200 SM до V3.0 - Запасные терминальные блоки

Имеется S7-1200 SM до V3.0 (заказной номер)	**В этом случае применяются следующие запасные терминальные блоки (4/упаковка)**	
	Заказной номер терминального блока	**Описание терминального блока**
SM 1221 DI 8 x DC (6ES7221-1BF30-0XB0)	6ES7292-1AG30-0XA0	7-полюсный терминальный блок с луженными контактами
SM 1222 DQ 8 x DC (6ES7222-1BF30-0XB0)		
SM 1222 DQ 8 x RLY (6ES7222-1HF30-0XB0)		
SM 1231 AI 4 x 13 бит (6ES7231-4HD30-0XB0)	6ES7292-1BG30-0XA0	7-полюсный терминальный блок с позолоченными контактами
SM 1232 AQ 2 x 14 бит (6ES7232-4HB30-0XB0)		
SM 1231 AI 4 x TC (6ES7231-5QD30-0XB0)		
SM 1231 AI 4 x 16 бит (6ES7231-5ND30-0XB0)		
SM 1221 DI 16 x DC (6ES7221-1BH30-0XB0)	6ES7292-1AG30-0XA0	7-полюсный терминальный блок с луженными контактами
SM 1222 DQ 16 x DC (6ES7222-1BH30-0XB0)		
SM 1222 DQ 16 x RLY (6ES7222-1HH30-0XB0)		
SM 1223 DI 8 x DC / DQ 8x DC (6ES7223-1BH30-0XB0)		
SM 1223 8 x DC / 8 x RLY (6ES7223-1PH30-0XB0)		
SM 1223 8 x AC / 8 x RLY (6ES7223-1QH30-0XB0)		
SM 1234 AI 4 / AQ 2 (6ES7234-4HE30-0XB0)	6ES7292-1BG30-0XA0	7-полюсный терминальный блок с позолоченными контактами
SM 1231 AI 8 x 13 бит (6ES7231-4HF30-0XB0)		
SM 1232 AQ 4 x 14 бит (6ES7232-4HD30-0XB0)		
SM 1231 AI 4 x RTD (6ES7231-5PD30-0XB0)		
SM 1231 AI 8 x TC (6ES7231-5QF30-0XB0)		
SM 1222 DQ 8 x RLY (переключение) (6ES7222-1XF30-0XB0)	6ES7292-1AL30-0XA0	11-полюсный терминальный блок с луженными контактами
SM 1223 DI 16 x DC/DQ 16 x DC (6ES7223-1BL30-0XB0)		
SM 1223 16 x DC/16 x RLY (6ES7223-1PL30-0XB0)		
SM 1231 AI 8 x RTD (6ES7231-5PF30-0XB0)	6ES7292-1BL30-0XA0	11-полюсный терминальный блок с позолоченными контактами

Указатель

-

-(RESET_BF) (сброс битового поля), 223

&

&-блок (логическая операция AND в FBD), 220

/

/=-блок (отрицание назначения в FBD), 221

=

=-блок (назначение FBD), 221

>

>=1-блок (логическая операция OR в FBD), 220

A

ABS (создание абсолютного значения), 251
AC
  Заземление, 76
  Правила электромонтажа, 74, 76
  Руководство по изоляции, 75
ACOS (определение значения арккосинуса), 254
ACT_TINT (активация прерывания по времени), 430
Ad hoc режим, TCP и ISO-on-TCP, 831
ADD (сложение), 248
ASi
  RDREC (чтение набора данных), 387
  WRREC (запись набора данных), 387
  Адрес, 994
  Ведущее устройство AS-i CM 1243-2, 990
  Добавление ведомого AS-i устройства, 992
  Добавление модуля ведущего AS-i устройства CM1243-2, 992
  Конфигурирование ведомого устройства без STEP 7, 996
  Конфигурирование ведомого устройства с помощью STEP 7, 998
  Назначение адресов ведомых устройств системой, 996
  Назначение системой, 996
  Передача аналоговых значений, 998
  Передача цифровых значений, 998
  Сетевое соединение, 992
AS-i
  Инструкции для распределенной периферии, 386
ASIN (определение значения арксинуса), 254
AT наложение для переменной (тега), 134
ATH (преобразование строки символов ASCII в шестнадцатеричное число), 364
ATTACH (назначить OB прерывающему событию), 421
ATTR_DB (чтение атрибутов блока данных), 544
AWP_Enum_Def, 1081
AWP_Import_Fragment, 1084
AWP_In_Variable, 1075, 1079
AWP_Out_Variable, 1077
AWP_Start_Fragment, 1083
AWP-команды, 1072
  Адресация типа перечисления, 1082
  Запись переменных, 1075
  Запись специальных переменных, 1079
  Импорт фрагментов, 1084
  Использование псевдонимов, 1080
  Объединение определений, 1085
  Определение типа перечисления, 1081
  Создание фрагментов, 1083
  Чтение специальных переменных, 1077

B

BB 1297, 1608

C

CALCULATE (вычисление), 247
CALCULATE (вычислить)
  Использование для сложных математических уравнений, 41
  Масштабировать аналоговые значение, 42
CAN_DINT (отмена прерывания по задержке времени), 432
CAN_TINT (удаление прерывания по времени), 430

CB 1241
Оконечная нагрузка, 1119
CB 1241 RS485, 1620
CEIL (округление числа с плавающей запятой до следующего целого числа в сторону увеличения), 304
Char (тип данных Character), 129
Chars_TO_Strg (преобразование массива из CHAR в строку символов), 362
CONCAT (объединение строк символов), 368
CONTINUE, SCL, 331
CONV (преобразование значения), 299
COS (определение значения косинуса), 254
CountOfElements (запрос количества элементов массива ARRAY), 276
CPU
ASi, 993
CPU 1211C AC/DC/RLY, 1444
CPU 1211C DC/DC/DC, 1444
CPU 1211C DC/DC/RLY, 1444
CPU 1212C AC/DC/RLY, 1457
CPU 1212C DC/DC/DC, 1457
CPU 1212C DC/DC/RLY, 1457
CPU 1214C AC/DC/RLY, 1470
CPU 1214C DC/DC/DC, 1470
CPU 1214C DC/DC/RLY, 1470
CPU 1215C AC/DC/RLY, 1484
CPU 1215C DC/DC/DC, 1484
CPU 1215C DC/DC/RLY, 1484
CPU 1217C DC/DC/DC, 1499
Ethernet порт, 819
IP-адрес, 819
MAC-адрес, 819, 824
PROFINET, 819
PROFINET IO, 940
RTM (счетчик часов работы), 350
Адрес AS-i, 994
Адрес PROFIBUS, 988
Баланс мощности, 55
Восстановление резервной копии, 1430
Время реакции на скачок, 1453, 1466, 1479, 1493, 1510
Выполнение программы, 83
Выход в онлайн, 1390
Добавление модулей, 156
Добавление нового устройства, 152
Загрузка в CPU, 209
Загрузка в устройство, 824
Заземление, 76
Защита ноу-хау, 206
Защита от несанкционированного доступа с помощью паролей, 203
Импульсные выходы, 492
Индуктивные нагрузки, 79
Количество коммуникационных соединений, 808
Коммуникационная нагрузка, 108
Коммуникационные платы (CB), 31
Коммуникация, 810
Конфигурация HSC, 574
Конфигурация устройства, 151
Конфигурирование времени цикла, 108
Копирование блоков из Онлайн-CPU, 214
Ламповые нагрузки, 78
Мониторинг онлайн, 1401
Монтаж, 59, 60
Назначение IP-адресов для CPU в режиме онлайн, 817
Не специфицированный CPU, 154
Обзор, 27
Обработка OB, 183
Обработка при запуске, 90
Онлайн, 1393
Панель управления, 47
Панель управления (Онлайн-CPU), 1397
Параметры запуска, 139
Поведение при перегрузке, 105
Порт AS-i, 993
Порядок действий при утерянном пароле, 149
Потребляемая мощность, 1633
Правила электромонтажа, 74, 76
Принудительное присваивание значений, 1408, 1409
Просмотр MAC- и IP-адресов, 824
Пустая трансферная карта, 149
Рабочие состояния, 87
Разрешение выходов в режиме STOP, 1407
Расширительный кабель, 68
Регистрация и сброс значений DB, 1403
Режимы работы RUN/STOP, 1397
Резервное копирование, 1428
Руководство по изоляции, 75
Сброс на заводские настройки, 1393
Светодиодная индикация, 1387
Свойство синхронизации времени, 826
Сетевое соединение, 811
Сигнальные платы (SB), 31
Совместимость версий, 50
Сравнение и синхронизация блоков, 1399
Сравнительная таблица, 28
Таблица текущего контроля, 1404
Тепловая зона, 54, 58
Терминальный блок, 67
Типы коммуникации, 805

Уровни защиты, 203
Утерянный пароль, 149
Экранные кнопки RUN/STOP, 47
CP-модуль
Доступ к веб-серверу, 1033
Начальная страница веб-сервера, 1042
CREATE_DB (создание блока данных), 537
CTD (обратный счёт), 236
CTRL_HSC (управление высокоскоростным счетчиком), 587
CTRL_HSC_EXT (управление высокоскоростным счетчиком (расширенное)), 558
CTS (аппаратное управление потоком, PtP), 1125
CTU (прямой счёт), 236
CTUD (прямой и обратный счёт), 236

D

D_ACT_DP, 399
DataLogClear, 520
DataLogDelete, 523
Date
DTL (тип данных Date and Time long), 128
Тип данных Date, 127
DB (блок данных), (Блок данных)
DB_ANY_TO_VARIANT (преобразование DB_Any в VARIANT), 309
DC
Выходы, 1441
Заземление, 76
Индуктивные нагрузки, 79
Правила электромонтажа, 74, 76
Руководство по изоляции, 75
DEC (уменьшение на 1), 251
DECO (декодирование), 335
DELETE (удаление символов из строки символов), 370
DELETE_DB (удаление блока данных), 546
DEMUX (демультиплексирование), 338
DETACH (отменить назначение OB прерывающему событию), 421
DeviceStates, пример, 469
DIV (деление), 248
DPNRM_DG, 417
DPRD_DAT (чтение непротиворечивых данных стандартного ведомого устройства DP), 409
DPWR_DAT (запись непротиворечивых данных стандартного ведомого устройства DP), 409

E

E-Mail, отправка с помощью TMAIL_C, 891
EN и ENO (поток сигналов), 201
EN_AIRT (активация выполнения прерываний с более высоким приоритетом и асинхронных ошибок), 435
ENCO (кодирование), 335
ENDIS_PW (ограничение и восстановление действия пароля), 316
EQ_ElemType (сравнить тип данных элемента МАССИВА на предмет НЕСОВПАДЕНИЯ с типом данных переменной (тега)), 245
EQ_ElemType (сравнить тип данных элемента МАССИВА на предмет СОВПАДЕНИЯ с типом данных переменной (тега)), 245
EQ_Type (сравнить тип данных на предмет СОВПАДЕНИЯ с типом данных переменной (тега)), 245
Ethernet
Ad hoc режим, 831
DPNRM_DG (чтение диагностических данных ведомого устройства DP), 417
DPRD_DAT (чтение непротиворечивых данных стандартного ведомого устройства DP), 409
DPWR_DAT (запись непротиворечивых данных стандартного ведомого устройства DP), 409
GET (считывание данных из удаленного CPU), 1002
IP-адрес, 819
MAC-адрес, 819
PRVREC (предоставление набора данных), 415
PUT (запись данных в удаленный CPU), 1002
RCVREC (получение набора данных), 412
RDREC (чтение набора данных), 387
T_CONFIG (настройка интерфейса), 919
TCON, 864
TDISCON, 864
TRCV, 864
TRCV_C, 843
TSEND, 864
TSEND_C, 843
TURCV (получение данных через Ethernet (UDP)), 914
TUSEND (передача данных через Ethernet (UDP)), 914
WRREC (запись набора данных), 387
Идентификаторы соединений, 832
Инструкция RALRM (получение прерывания), 395

Количество коммуникационных соединений, 808
Компактный коммутационный модуль CSM 1277, 1628
Обзор, 828
Сетевое соединение, 811
Старая инструкция TRCV_C (получение данных через Ethernet (TCP)), 856
Старая инструкция TSEND_C (передача данных через Ethernet (TCP)), 856
Старые инструкции TCON, TDISCON, TSEND и TRCV, 876
Типы коммуникации, 805
Ethernet протоколы, 828
Соединения с несколькими узлами, 1006
EXIT, SCL, 332
EXP (определение экспоненциального значения), 254
EXPT (возведение в степень), 254

F

F_TRIG (установка переменной (тега) по отрицательному фронту сигнала), 226
FB (функциональный блок)
Обзор, 83
FBD (функциональная блок-схема), 193
FC (функция), 83, 184
FieldRead (чтение поля), 277
FieldWrite (запись поля), 277
FILL_BLK (заполнить блок), 265
FIND (поиск символов в строке символов), 373
FLOOR (округление числа с плавающей запятой до следующего целого числа в сторону уменьшения), 304
FOR, SCL, 329
FRAC (определение количества разрядов после запятой), 254

G

Gen_UsrMsg (создание диагностических сообщений пользователя), 436
GEOADDR, 553
GET (считывание данных из удаленного CPU), 1002
Конфигурирование соединения, 813
GET_DIAG (чтение информации системы диагностики), 479
GET_ERROR (локальное получение ошибки), 320
GET_ERROR_ID (локальное получение ID ошибки), 321
GetBlockName (считывание имени блока), 384
GetInstanceName (считывание имени экземпляра блока), 380
GETIO, 390
GETIO_PART, 392
GetStationInfo, 458
GetSymbolName (считывание имени переменной во входном параметре), 374
GetSymbolPath (запрос составного глобального имени входного параметра), 377
GetSymbolPath (запрос составного глобального имени экземпляра блока), 382
GOTO, SCL, 333

H

HSC (высокоскоростные счетчики)
Конфигурация, 574
Рабочая фаза, 576
Типы счета, 575
HTA (преобразование шестнадцатеричного числа в строку символов ASCII), 364
HTML-страницы
Определяемые пользователем, 1071
Список, пример для пользовательских веб-страниц, 1099
HTML-страницы, пользовательские
Доступ к данным S7-1200, 1072
Места хранения для языков, 1108
Обновление, 1072
Организация страниц, 1087
Разработка, 1071
HTTP-соединения, веб-сервер, 1112

I

I/O
Адресация, 121
Аналоговая индикация состояния, 1389, 1390
Время реакции на скачок (CPU), 1453, 1466, 1479, 1493, 1510
Время реакции на скачок (SB), 1594
Время реакции на скачок (SM), 1547
Индуктивные нагрузки, 79
Мониторинг в таблице текущего контроля, 1404
Мониторинг состояния в LAD, 1402
Представление аналогового входа (напряжение), 1548, 1594

Представление аналогового входа (ток), 1548, 1595
Представление аналогового выхода (напряжение), 1549, 1596
Представление аналогового выхода (ток), 1550, 1596
Функция принудительной установки, 1409
Цифровая индикация состояния, 1388
IF-THEN, SCL, 326
IN_Range (значение в диапазоне), 243
INC (увеличение на 1), 251
INSERT (вставка символов в строку символов), 371
INV (создание дополнения до единицы), 335
IO2MOD (определение аппаратного идентификатора по адресу ввода/вывода), 551
IO-Link
Возврат к заводским настройкам, 1568
Диагностика, 1576
Диаграмма, 1570
Диапазон адресов, 1571
Замена, 1568
Изменение параметров во время работы, 1572
Конфигурирование, 1571
Набор данных, 1572
Параметры, 1571
Профиль устройства, 1566
Разводка контактов, 1569
Светодиодный индикатор, 1574
Сообщения об ошибках, 1572, 1574, 1576
Функции, 1568
Хранение устройства, 1569
IO-Link-Master сигнальный модуль, 1563
IP-адрес, 819, 820
MAC-адрес, 819
Конфигурация устройства, 167
Конфигурирование, 819
Конфигурирование CPU онлайн, 1393
Назначение, 815, 823
Присвоение в режиме онлайн, 817
IP-адрес маршрутизатора, 820
IP-адрес, аварийный адрес (временный), 1016
IP-маршрутизатор, 819
IS_ARRAY (проверка на МАССИВ), 246
IS_NULL (запрос на РАВЕНСТВО указателя НУЛЮ), 246
ISO on TCP
Ad hoc режим, 831
ISO-on-TCP
Идентификаторы соединений, 832
Конфигурация соединения, 812
Параметр, 835
I-устройство (интеллектуальное IO устройство)
Конфигурирование, 950
Конфигурирование с помощью файла GSD, 952
общего доступа, 961
I-устройство (интеллектуальное устройство ввода-вывода)
низкоуровневая PN IO-система, 944
Свойства, 944
I-устройство общего доступа, конфигурация, 961

J

JavaScript, стандартные веб-страницы, 1113
JMP (переход при RLO = 1), 311
JMP_LIST (определение списка переходов), 312
JMPN (переход при RLO = 0), 311

L

Label (метка перехода), 311
LAD (релейно–контактная схема)
Мониторинг, 1402
Обзор, 192
Редактор текстов программы, 1402
Состояние, 1402, 1408
Состояние или значение мониторинга, 1401
LED (чтение состояния светодиода), 449
LEFT (считывание левых символов в строке символов), 369
LEN (определение длины строки символов), 367
LIMIT (установка предельного значения), 253
LN (определение натурального логарифма), 254
LOG2GEO (определение слота по аппаратному идентификатору), 549
LOWER_BOUND (считывание нижней границы массива ARRAY), 268

M

MAC-адрес, 819, 824
MAX (вычислить максимум), 252
MAX_LEN (максимальная длина строки символов), 367
MB_CLIENT, 1200
MB_CLIENT (обмен данными с использованием PROFINET в качестве клиента Modbus TCP), устаревшая, 1329

MB_COMM_LOAD (конфигурирование порта на PtP модуле для Modbus RTU), устаревшая, 1347
MB_MASTER (обмен данными через PtP-порт в качестве ведущего устройства Modbus), устаревшая, 1350
MB_RED_CLIENT, 1221
MB_RED_SERVER, 1240
MB_SERVER, 1209
MB_SERVER (обмен данными с использованием PROFINET в качестве сервера Modbus TCP), устаревшая, 1336
MB_SLAVE (обмен данными через PtP-порт в качестве ведомого устройства Modbus), устаревшая, 1356
MC_CommandTable, 756
MC_MoveAbsolute (абсолютное позиционирование оси), 746
MID (считывание средних символов в строке символов), 369
MIN (вычислить минимум), 252
MOD (остаток от деления), 249
Modbus
  MB_CLIENT (обмен данными с использованием PROFINET в качестве клиента Modbus TCP), устаревшая, 1329
  MB_COMM_LOAD (конфигурирование порта на PtP модуле для Modbus RTU), устаревшая, 1347
  MB_MASTER (обмен данными через PtP-порт в качестве ведущего устройства Modbus), устаревшая, 1350
  MB_SERVER (обмен данными с использованием PROFINET в качестве сервера Modbus TCP), устаревшая, 1336
  MB_SLAVE (обмен данными через PtP-порт в качестве ведомого устройства Modbus), устаревшая, 1356
  Modbus_Master (обмен данными через SIPLUS I/O или PtP-порт в качестве ведущего устройства Modbus RTU), 1274
  Modbus_Slave (обмен данными через SIPLUS I/O или PtP-порт в качестве ведомого устройства Modbus RTU), 1282
  RTU коммуникация, 1197
  Адреса в памяти, 1197
  Адреса станции в сети, 1197
  Версии, 44, 1179, 1269, 1315, 1346
  Инструкция Modbus_Comm_Load (конфигурирование SIPLUS I/O или порта на PtP модуле для Modbus RTU), 1270
  Функциональные коды, 1195
MODBUS
  MB_CLIENT, 1200
  MB_RED_CLIENT, 1221
  MB_RED_SERVER, 1240
  MB_SERVER, 1209
Modbus RTU
  Пример ведомого устройства, 1296
  Программа для ведущего устройства, 1293
Modbus TCP
  Версии, 1199, 1328
Modbus_Master (обмен данными через SIPLUS I/O или PtP-порт в качестве ведущего устройства Modbus RTU), 1274
Modbus_Slave (обмен данными через SIPLUS I/O или PtP-порт в качестве ведомого устройства Modbus RTU), 1282
ModuleStates, 474
ModuleStates пример, 475
MOVE (копирование значения), 256
MOVE_BLK (копирование области), 256
MRES, панель управления, 47
MUL (умножение), 248
MUX (мультиплексирование), 337

N

N (опрос операнда на отрицательный фронт сигнала), 224
N_TRIG (опрос RLO на отрицательный фронт сигнала), 225
N=-блок и N-катушка (установка операнда при отрицательном фронте сигнала), 225
NE_Type (сравнить тип данных на предмет НЕСОВПАДЕНИЯ с типом данных переменной (тега)), 245
NEG (создать двоичное дополнение), 250
Network Time Protocol (NTP), 826
NO-/NC-катушка, 221
NO-/NC-контакт, 219
NORM_X (нормализация), 305
NOT (НЕ) (инверсия RLO), 221
NOT_NULL (запрос на ОТЛИЧИЕ указателя от НУЛЯ), 246
NOT_OK (проверить на недостоверность), 244

O

OB, (Организационный блок)
OB "Удаление или вставка модулей", 98
OB MC-PostServo, 102
OB MC-PreServo, 102
OB аппаратных прерываний, 94
OB диагностического прерывания, 96

OB ошибки стойки или станции ("Rack or station failure OB"), 99
OB прерывания по времени, 100
OB прерывания по задержке времени, 92
OB прерывания по обновлению, 101
OB прерывания по профилю, 101
OB программного цикла, 91
OB состояния, 100
OK (проверить на достоверность), 244
OPC, конфигурирование, 1375
OUT_Range (значение вне диапазона), 243

P

P (опрос операнда на положительный фронт сигнала), 224
P_TRIG (опрос RLO на положительный фронт сигнала), 225
P=-блок и P-катушка (установка операнда при положительном фронте сигнала), 225
P3964_Config (конфигурирование протокола 3964(R)), 1152
  Error, 1153
PEEK, PEEK_WORD, PEEK_BOOL, PEEK_DWORD, PEEK_BLK, 199, 271
PLC
  RTM (счетчик часов работы), 350
  Баланс мощности, 55
  Время цикла, 107, 108
  Добавление модулей, 156
  Загрузка в CPU, 209
  Защита ноу-хау, 206
  Коммуникационная нагрузка, 108
  Конфигурация HSC, 574
  Конфигурация устройства, 151
  Копирование блоков из Онлайн-CPU, 214
  Мониторинг, 1401
  Монтаж, 59, 60
  Назначение IP-адресов для CPU в режиме онлайн, 817
  Обзор CPU, 27
  Обработка при запуске, 90
  Переменные (теги), 116
  Принудительное присваивание значений, 1408
  Рабочие состояния, 87
  Разработка решения автоматизации, 179
  Расширительный кабель, 68
  Свойство синхронизации времени, 826
  Сравнение и синхронизация, 1399
  Таблица текущего контроля, 1404
  Терминальный блок, 67
  Функция принудительной установки, 1409
POKE, POKE_BOOL, POKE_BLK, 199, 271
PORT_CFG (динамическое конфигурирование коммуникационных параметров), устаревшая, 1298
Port_Config (конфигурирование порта), 1141
PROFIBUS
  CM 1242-5 (ведомое устройство DP), 984
  CM 1243-5 (ведущее устройство DP), 984
  DPNRM_DG (чтение диагностических данных ведомого устройства DP), 417
  DPRD_DAT (чтение непротиворечивых данных стандартного ведомого устройства DP), 409
  DPWR_DAT (запись непротиворечивых данных стандартного ведомого устройства DP), 409
  GET (считывание данных из удаленного CPU), 1002
  PUT (запись данных в удаленный CPU), 1002
  RDREC (чтение набора данных), 387
  S7-соединение, 1006
  WRREC (запись набора данных), 387
  Адрес, 988
  Ведомое устройство, 984
  Ведущее устройство, 984
  Добавление CM 1243-5 (ведущее устройство DP), 987
  Добавление ведомого устройства DP, 987
  Инструкции для распределенной периферии, 386
  Инструкция RALRM (получение прерывания), 395
  Количество коммуникационных соединений, 808
  Конфигурирование адреса, 988
  Сетевое соединение, 811, 988
PROFIBUS и PROFINET
  ModuleStates пример, 475
  Пример DeviceStates, 469
PROFIdrive, 676
PROFIenergy, 420
PROFINET
  Ad hoc режим, 831
  DPRD_DAT (чтение непротиворечивых данных стандартного ведомого устройства DP), 409
  DPWR_DAT (запись непротиворечивых данных стандартного ведомого устройства DP), 409
  GET (считывание данных из удаленного CPU), 1002
  IP-адрес, 819
  MAC-адрес, 819
  PRVREC (предоставление набора данных), 415

PUT (запись данных в удаленный CPU), 1002
RCVREC (получение набора данных), 412
RDREC (чтение набора данных), 387
S7-соединение, 1006
WRREC (запись набора данных), 387
Идентификаторы соединений, 832
Имена устройств и адресация, 828
Инструкции для распределенной периферии, 386
Инструкция RALRM (получение прерывания), 395
Количество коммуникационных соединений, 808
Коммуникация PLC/PLC, 935
Конфигурирование IP-адреса, 167
Конфигурирование коммуникации между CPU и устройством HMI, 934
Межпроцессорная коммуникация, 935
Назначение IP-адреса, 828
Начальное время системы, 827
Обзор, 828
Разорвать и после восстановить существующее соединение, 885
Свойства Ethernet-адреса, 820
Свойство синхронизации времени, 826
Сетевое соединение, 811, 935, 936, 940
Синхронизация времени, 167
Тестирование сети, 823
Типы коммуникации, 805
PROFINET IO
Добавление устройства, 940
Имена устройств, 941
Назначение CPU, 941
Онлайн имена устройств, 1391
Присвоение имен устройствам, 941
Присвоение имен устройствам онлайн, 1391
Устройства, 940
PROFINET RT, 828
PROFINET порт
Автоопределение, 822
PRVREC (предоставление набора данных), 415
PTO, 666
PTO (вывод последовательности импульсов)
CTRL_PTO (вывод последовательности импульсов), 489
CTRL_PWM (широтно-импульсная модуляция), 487
Конфигурирование импульсных каналов, 494
Принудительная установка значений не может быть выполнена, 1409
Принцип работы, 492
PtP-коммуникация, 1120
Конфигурирование параметров, 1126
Конфигурирование портов, 1123
Конфигурирование программы-примера, 1168
Пример программы, 1167
Программа-пример, выполнение, 1176
Программа-пример, программирование STEP 7, 1174
Программирование, 1165
Эмулятор терминала для программы-примера, 1175
PtP-коммуникация, 3964(R)
Конфигурирование портов, 1136
Конфигурирование приоритета и параметров протокола, 1137
PUT (запись данных в удаленный CPU), 1002
Конфигурирование соединения, 813
PWM (широтно-импульсная модуляция)
CTRL_PTO (вывод последовательности импульсов), 489
CTRL_PWM (широтно-импульсная модуляция), 487
Адреса ввода-вывода, 497
Время цикла, 495
Длительность импульса, 495
Изменение времени цикла, 497
Изменение длительности импульса, 497
Конфигурирование импульсных каналов, 494
Принудительная установка значений не может быть выполнена, 1409
Принцип работы, 492

Q

QRY_CINT (запрос параметров циклического прерывания), 426
QRY_DINT (запрос состояния прерывания по задержке времени), 432
QRY_TINT (запрос состояния прерывания по времени), 431

R

R (сброс выхода), 222
R_TRIG (установка переменной (тега) по положительному фронту сигнала), 226
RALRM (получение прерывания), 404
RCV_CFG (динамическое конфигурирование параметров последовательного приема), устаревшая, 1301

RCV_PTP (разрешить прием сообщений), 1309
RCV_RST (удалить данные в принимающем буфере), устаревшая, 1311
RCVREC (получение набора данных), 412
RD_ADDR (определение адреса ввода/вывода по аппаратному идентификатору), 552
RD_LOC_T (считывание местного времени), 346
RD_SINFO (считывание начальной информации текущего блока OB), 440
RD_SYS_T (считывание реального времени), 346
RDREC (чтение набора данных), 387, 404
RE_TRIGR (перезапуск времени контроля цикла), 318
READ_BIG (чтение данных в формате Big-Endian), 272
READ_DBL (чтение из блока данных в загружаемой памяти), 541
READ_LITTLE (чтение данных в формате Little-Endian), 272
Receive_Config (конфигурация приема), 1147
Receive_P2P (прием данных по коммуникации "точка-точка"), 1158
REPEAT, SCL, 331
REPLACE (замена символов в строке символов), 372
RET (возвращаемое значение), 315
RETURN, SCL, 333
RIGHT (считывание правых символов в строке символов), 369
ROL (циклический сдвиг влево) и ROR (циклический сдвиг вправо), 341
ROUND (округление числа), 303
RS (триггер сброса/установки), 223
RT (сброс таймера), 227
RTS (аппаратное управление потоком, PtP), 1125
RUNTIME (измерение времени выполнения программы), 323

S

S (установка выхода), 222
S_CONV (преобразование строки символов), 353
S_MOV (перемещение строки символов), 353
S7-маршрутизация, 978
SCALE_X (масштабирование), 305
SCL (Structured Control Language)
  EN и ENO (поток сигналов), 201
  Адресация, 196
  Битовая логика, 219
  Вызов FB или FC, 196
  Вызов блоков, 182
  Выражения, 196
  Инструкции преобразования, 300
  Обзор, 194
  Операторы, 196
  Приоритет операторов, 196
  Раздел Var, 194
  Редактор текстов программ, 194
  Сравнение значений, 242
  Таймеры, 227
  Управление программой, 325
  Управляющие операторы, 196, 325
  Условия, 196
SEL (выбор), 337
SEND_CFG (динамическое конфигурирование параметров последовательной передачи), устаревшая, 1300
Send_Config (конфигурация передачи), 1144
Send_P2P (передача данных в рамках коммуникации "точка-точка"), 1154
  Параметры LENGH и BUFFER, 1157
SEND_PTP (отправить данные из передающего буфера), устаревшая, 1307
SET_BF (установка битового поля), 223
SET_CINT (параметрирование циклического прерывания), 424
Set_Features (установка расширенных функций), 1164
SET_TIMEZONE (установка часового пояса), 349
SET_TINTL (установка даты и времени для прерывания по времени), 428
SETIO, 391
SETIO_PART, 394
SGN_GET (запросить сигналы RS232), устаревшая, 1312
SGN_GET (получить сигналы RS232), 1161
SGN_SET (установить сигналы RS-232), устаревшая, 1313
SHL (сдвиг влево) и SHR (сдвиг вправо), 340
Signal_Set (установка сигналов RS232), 1162
SIN (определение значения синуса), 254
SM 1231 RTD
  Таблицы выбора, 1560, 1605
SM и SB
  Конфигурация устройства, 151
  Сравнительная таблица, 31
SMS, 1374
SQR (определение квадрата), 254
SQRT (определение квадратного корня), 254
SR (триггер сброса/установки), 223
SRT_DINT (запуск прерывания по задержке времени), 432
STARTUP, рабочее состояние
  Функция принудительной установки, 1409

STEP 7
ASi, 994
Ethernet порт, 819
PROFIBUS, 988
PROFINET, 819
RTM (счетчик часов работы), 350
Блок данных (DB), 83
Блок данных экземпляра (DB), 185
Время цикла, 107, 108
Вставка инструкций, 40
Вызов блоков кода в программе пользователя, 182
Вызовы блоков, 83
Добавление PROFINET IO-устройства, 940
Добавление входов или выходов в LAD или FBD инструкцию, 43
Добавление модулей, 156
Добавление нового устройства, 152
Загрузка в CPU, 209
Защита паролем, 206
Избранное, 41
Изменение настроек, 45
Класс приоритетов (OB), 91
Количество блоков кода, 83
Коммуникационная нагрузка, 108
Конфигурация HSC, 574
Конфигурация устройства, 151
Конфигурирование CPU, 167
Конфигурирование модулей, 174
Копирование блоков из Онлайн-CPU, 214
Линейные и структурированные программы, 180
Мониторинг, 1401, 1402
Назначение IP-адресов для CPU в режиме онлайн, 817
Начальное значение FB, 185
Не вставленные модули, 50
Обработка при запуске, 90
Панель управления, 47
Перетаскивание между редакторами, 46
Порт AS-i, 993
Портальное представление и проектное представление, 39
Правильные номера FC, FB и DB, 83
Принудительное присваивание значений, 1408
Принцип работы, 1404
Программная карта, 136
Рабочие состояния, 87
Расширяемые входы и выходы, 44
Свойство синхронизации времени (PROFINET), 826
Сетевое соединение, 811
Совместимость версий, 50
Сравнение и синхронизация, 1399
Функциональный блок (FB), 83, 185
Функция (FC), 184
Функция принудительной установки, 1409
Экранные кнопки RUN/STOP, 46
STP (завершение программы), 319
Strg_TO_Chars (преобразование строки символов в массив из CHAR), 362
STRG_VAL (преобразование строки символов в числовое значение), 353
String
S_MOV (перемещение строки символов), 353
Обзор строковых операций, 366
Тип данных STRING, 129
Тип данных String (строка), 352
SUB (вычитание), 248
SWAP (поменять месторасположение), 267
SWITCH (распределение переходов), 313

T

T_ADD (суммирование времени), 344
T_COMBINE (объединение времени), 345
T_CONFIG (настройка интерфейса), 919
T_CONV (преобразование и извлечение значений времени), 343
T_DIAG, 887
T_DIFF (разность времен), 345
T_RESET, 885
T_SUB (вычитание времени), 344
TAN (определение значения тангенса), 254
TCON, 864
Идентификаторы соединений, 832
Конфигурация, 812
Параметры соединения, 835
TCON, TDISCON, TSEND и TRCV
Версии, 863, 875
TCON_Param, 835
TCP
Ad hoc режим, 831
Идентификаторы соединений, 832
Конфигурация соединения, 812
Параметр, 835
Протоколы, 828
TDISCON, 864
TeleService адаптер и модуль, 70
TeleService через GPRS, 1371
TIA Portal, портальное представление и проектное представление, 39

Time
DTL (тип данных Date and Time long), 128
TOD (тип данных Time of Day), 127
Тип данных Time, 127
TimeTransformationRule для летнего времени, 348
TM_Mail (передача электронного сообщения), 1379
TMAIL_C, 891
TRCV, 864
Идентификаторы соединений, 832
TRCV (получение данных через Ethernet (TCP))
Ad hoc режим, 831
Конфигурация параметров, 939
TRCV_C
Ad hoc режим, 831
TRCV_C (получение данных через Ethernet (TCP)), 843
Идентификаторы соединений, 832
Конфигурация, 812
Параметры соединения, 835
TRUNC (округление до целого числа), 303
TS адаптер
SIM карта, 71
Монтаж TS модуля, 70
Монтаж на DIN-рейку, 72
Настенный монтаж, 73
TSAP (Transportation Service Access Point), 814
Запрещенные TSAP и номера портов, 931
Инструкции для назначения устройствам, 828
Конфигурирование общих параметров, 937, 1007
Определение, 830
TSEND, 864
Идентификаторы соединений, 832
TSEND_C (передача данных через Ethernet (TCP)), 843
Идентификаторы соединений, 832
Конфигурация, 812
Конфигурация инструкции, 938
Параметры соединения, 835
TSEND_C и TRCV_C
Версии, 843
Старые версии, 856
TURCV (получение данных через Ethernet (UDP)), 914
Конфигурация, 812
Параметры соединения, 835
TUSEND (передача данных через Ethernet (UDP)), 914
Конфигурация, 812
Параметр, 835

U

UDP
Конфигурация соединения, 812
Параметр, 835
UFILL_BLK (заполнить блок без прерываний), 265
UMOVE_BLK (непрерывное копирование области), 256
UPPER_BOUND (считывание верхней границы массива ARRAY), 269

V

VAL_STRG (преобразование числового значения в строку символов), 353
VARIANT_TO_DB_ANY (преобразование из VARIANT в DB_ANY), 308
VariantGet (чтение значения переменной (тега) VARIANT), 274
VariantPut (запись значения переменной (тега) VARIANT), 275

W

WChar (тип данных Word Character), 129
WHILE, SCL, 330
WR_LOC_T (установка местного времени), 346
WR_SYS_T (установка реального времени), 346
WRIT_DBL (запись в блок данных в загружаемой памяти), 541
WRITE_BIG (запись данных в формате Big-Endian), 272
WRITE_LITTLE (запись данных в формате Little-Endian), 272
WRREC (запись набора данных), 387, 404
WString (тип данных Word String), 129
WWW (синхронизация пользовательских веб-страниц), 1089

X

XON / XOFF, 1126
XOR (ИСКЛЮЧАЮЩЕЕ ИЛИ) (логическая операция), 334
x-блок (логическая операция XOR в FBD), 220

А

Австралия и Новая Зеландия - апробация RCM Mark, 1436
Автоопределение, 822
Адаптер TS-Adapter, 31
Адреса
  Чтение MAC-адреса с помощью GetStationInfo, 458
  Чтение адреса устройства с помощью GetStationInfo, 458
Адреса в памяти, 116, 118
Адресация
  Логические значения или битовые значения, 117
  Области памяти, 116
  Образ процесса, 116
  Отдельные входы (I) или выходы (Q), 117
Активная/пассивная коммуникация
  Идентификаторы соединений, 832
  Конфигурирование участников, 812, 1008
  Параметр, 835
Активное/пассивное соединение, 812
Аналоговые I/O
  Время реакции на скачок (CPU), 1453, 1466, 1479, 1493, 1510
  Время реакции на скачок (SB), 1594
  Время реакции на скачок (SM), 1547
  Индикация состояния, 1389, 1390
  Конфигурация, 174
  Представление входа (напряжение), 1548, 1594
  Представление входа (ток), 1548, 1595
  Представление выхода (напряжение), 1549, 1596
  Представление выхода (ток), 1550, 1596
  Преобразование в физические единицы, 42, 122, 306
Аналоговые сигнальные модули
  SM 1231, 1537
  SM 1231 RTD, 1557
  SM 1231 Thermocouple (TC), 1550
  SM 1232, 1541
  SM 1234, 1543
Аналоговые сигнальные платы
  SB 1231, 1590
  SB 1231 RTD, 1602
  SB 1231 Thermocouple, 1597
  SB 1232, 1592
Аналоговый привод, 676
Аппаратное управление потоком, 1125
Арифметика, 41, 247, 248
Архив данных
  DataLogClose (закрытие архива данных), 522
  DataLogCreate (создание архива данных), 512
  DataLogNewFile (архив данных в новом файле), 524
  DataLogOpen (открыть архив данных), 516
  DataLogWrite (запись архива данных), 518
  Обзор архивов данных, 510
  Ограничение и расчет размера, 528
  Очистить с помощью DataLogClear, 520
  Пример программы, 532
  Просмотр архивов данных, 527
  Структура набора данных, 511
  Удалить с помощью DataLogDelete, 523
Архитектура опроса, 1166
Архитектура опроса ведомого устройства, 1167
Архитектура опроса ведущего устройства, 1166

Б

Базовые панели (HMI), 32
Байт системной памяти, 112
Байт тактовой памяти, 113
Баланс мощностей
  Обзор, 1633
  Пример, 1635
  Форма для вычисления, 1636
Баланс мощности, 55
Батарейная плата (BB)
  BB 1297, 1608
  Установка батареи, 1609
Без перезапуска, 87
Безопасность
  Защита от копирования, 207
  Привязка к CPU, карте памяти или паролю, 207
  Утерянный пароль, 149
Библиотека для протокола USS
  USS_Port_Scan (обработка коммуникации через сеть USS), 1182
  USS_Read_Param (считывание параметров из привода), 1187
  Инструкция USS_Drive_Control (обмен данными с приводом), 1184
  Инструкция USS_Write_Param (изменить параметры в приводе), 1188
  Коды состояния, 1190
  Необходимое условие для использования, 1180
  Обзор, 1176
Библиотека для старого протокола USS
  USS_DRV (обмен данными с приводом), 1320

USS_PORT (обработка коммуникации через сеть USS), 1318
USS_RPM (считывание параметров из привода), 1323
USS_WPM (изменение параметров в приводе), 1324
Коды состояния, 1326
Необходимое условие для использования, 1316
Обзор, 1314
Битовая логика
Нормально разомкнутые (NO) и нормально замкнутые катушки (NC), 221
Нормально разомкнутые (NO) и нормально замкнутые контакты (NC), 219
Операции AND (И), OR (ИЛИ) и XOR (ИСКЛЮЧАЮЩЕЕ ИЛИ), 220
Операции по положительному и отрицательному фронту, 225
Операции установки и сброса, 222
Операция NOT (НЕ) (инверсия RLO), 221
Биты захвата импульсов, конфигурация цифровых входов, 170
Блок данных
CONF_DATA, 922
READ_DBL (чтение из блока данных в загружаемой памяти), 541
WRIT_DBL (запись в блок данных в загружаемой памяти), 541
Блок данных экземпляра, 116
Глобальный блок данных, 116, 186
Импорт фрагментов на пользовательские веб-страницы, 1084
Обзор, 83, 186
Одиночный FB с мультиэкземплярными DB, 186
Оптимизированный доступ, 187
Организационные блоки (OB), 183
Регистрация и сброс значений, 1403
Синхронизация стартовых значений Онлайн- и Офлайн-CPU, 212
Создать с CREATE_DB, 537
Стандартный доступ, 187
Структура, 83
Удалить с помощью DELETE_DB, 546
Чтение атрибутов с помощью ATTR_DB, 544
Блок данных экземпляра, 116
Блок кода
DB (блок данных), 83, 186
FB (функциональный блок), 83, 185
FC (функция), 83, 184
Блок данных экземпляра (DB), 185
Вызов блоков кода в программе пользователя, 182
Вызовы блоков, 83
Глубина вложенности, 30, 1446, 1459, 1472, 1486, 1501
Защита ноу-хау, 206
Защита от копирования, 207
Количество блоков кода, 30, 83, 1446, 1459, 1472, 1486, 1501
Количество OB, 30, 1447, 1460, 1473, 1487, 1502
Контроль, 30
Линейные и структурированные программы, 180
Мониторинг, 1446, 1459, 1472, 1486, 1501
Начальное значение FB, 185
Организационные блоки (OB), 30, 183, 1447, 1460, 1473, 1487, 1502
Правильные номера FC, FB и DB, 83
Прерывания, 30, 1447, 1460, 1473, 1487, 1502
Привязка к CPU, карте памяти или паролю, 207
Размер программы пользователя, 30, 1446, 1459, 1472, 1486, 1501
Счётчики (количество и требуемая память), 30, 1447, 1460, 1473, 1487, 1502
Таймеры (количество и требуемая память), 30, 1447, 1460, 1473, 1487, 1502
Блок передачи (T-блок), 937
Блок питания
PM1207, 1628
Блок питания PM 1207, 1628
Блоки
Блок данных (DB), 83
Блок данных экземпляра (DB), 185
Вызов FB или FC с SCL, 196
Вызовы блоков, 83
Глубина вложенности, 30, 83, 1446, 1459, 1472, 1486, 1501
Загрузка, 209
Защита паролем, 206
Количество блоков кода, 30, 83, 1446, 1459, 1472, 1486, 1501
Количество OB, 30, 103, 1447, 1460, 1473, 1487, 1502
Копирование блоков из Онлайн-CPU, 214
Линейные и структурированные программы, 180
Мониторинг, 30, 1446, 1459, 1472, 1486, 1501
Начальное значение FB, 185
Организационные блоки (OB), 30, 83, 91, 103, 1447, 1460, 1473, 1487, 1502
Правильные номера FC, FB и DB, 83

Прерывания, 30, 103, 1447, 1460, 1473, 1487, 1502
Проверка непротиворечивости, 217
Пусковые OB, 103
Размер программы пользователя, 30, 83, 1446, 1459, 1472, 1486, 1501
События, 103
Счётчики (количество и требуемая память), 30, 1447, 1460, 1473, 1487, 1502
Таймеры (количество и требуемая память), 30, 1447, 1460, 1473, 1487, 1502
Типы, 83
Функциональный блок (FB), 83, 185
Функция (FC), 83, 184
Экземплярный или мультиэкземплярный DB, 185
Блоки обработки данных (DHB), 186
Блоки схемной защиты для индуктивных нагрузок, 79

B

Ввод в эксплуатацию
Инструкции PID_Compact и PID_3Step, 644
Инструкция PID_Temp, 646
Веб-сервер
Активация, 1026
Доступ мобильного устройства, 1031
Доступ через CP модуль, 1033
Конфигурирование пользователей, 1028
максимальное количество HTTP соединений, 1112
Ограничения, 1112
Поддерживаемые браузеры, 1025
Пользовательские веб-страницы, 1071
Правила для кавычек, 1085
Представление на мобильном устройстве, 1037
Скорость обновления, 1026
Стандартные веб-страницы, 1029
Веб-страница "Состояние переменных (тегов)", 1055
Веб-страницы
Сервис, поддержка и документация по STEP 7, 4
Веб-страницы STEP 7, 4
Ведомое устройство PN
Активация и деактивация с помощью D_ACT_DP, 399
Версии инструкций, 44, 843, 856, 863, 875, 1179, 1199, 1269, 1315, 1328, 1346
Вид защиты, 1440
Визуализация, устройства HMI, 32
Возвращаемые значения
Инструкции PtP, 1139
Инструкции для открытых коммуникационных соединений пользователя, 930
Возвращаемые значения инструкций для открытых коммуникационных соединений пользователя, 930
Возвращаемые значения, время выполнения приема, 1158, 1309
Восстановление резервной копии, 1430
Временная память
Использование блоками, 120
Максимум на каждый уровень приоритета OB, 120
Время
RD_LOC_T (считывание местного времени), 346
RD_SYS_T (считывание реального времени), 346
SET_TIMEZONE (установка часового пояса), 349
T_ADD (суммирование времени), 344
T_COMBINE (объединение времени), 345
T_CONV (преобразование и извлечение значений времени), 343
T_DIFF (разность времен), 345
T_SUB (вычитание времени), 344
WR_LOC_T (установка местного времени), 346
WR_SYS_T (установка реального времени), 346
Время ожидания, 1124
Время суток
Конфигурирование CPU онлайн, 1393
Время фильтра, 169
Время фильтра для входа, 169
Время фильтра для цифрового входа, 169
Время цикла
Конфигурация, 108
Мониторинг, 1397
Обзор, 107
Вставка инструкций
Избранное, 41
Перетаскивание, 40
Перетаскивание между редакторами, 46
Вставка карты памяти в CPU, 137
Вставка устройства
Не специфицированный CPU, 154
Входы
Биты захвата импульсов, 170
Входы в систему/выход из системы, стандартные веб-страницы, 1039

Входы и выходы
Мониторинг, 1401
Выгрузка из CPU
Копирование блоков из Онлайн-CPU, 214
Программа пользователя, 214
Вызов блоков кода в программе пользователя, 182
Выполнение команд управления осью как последовательности движений (MC_CommandTable), 756
Выполнение программы, 83
Высокоскоростной счетчик, 587
Принудительная установка значений не может быть выполнена, 1409
Высокоскоростные счетчики, 558
Конфигурация, 574
Рабочая фаза, 576
Типы счета, 575
Выходные параметры, 185
Импульсные выходы, 492
Конфигурирование импульсных каналов, 494

Г

Генерация DB для пользовательских веб-страниц, 1088
Гибкое оборудование (управление конфигурацией), 157
Глобальная библиотека
Обзор протокола USS, 1176
Обзор старого протокола USS, 1314
Глобальный блок данных, 116, 186
Глубина вложенности, 83
Горячая линия, 3
Горячий пуск, 87

Д

Дата
SET_TIMEZONE (установка часового пояса), 349
T_ADD (суммирование времени), 344
T_COMBINE (объединение времени), 345
T_CONV (преобразование и извлечение значений времени), 343
T_DIFF (разность времен), 345
T_SUB (вычитание времени), 344
Десериализация, 260
Диагностика
GET_DIAG (чтение информации системы диагностики), 479
LED (чтение состояния светодиода), 449
Буфер, 113
Буфер диагностики, 1397
Время цикла, 1397
Индикация состояния, 112
Инструкция DeviceStates (считывание состояния для модуля системы ввода/вывода), 468
Инструкция Get_IM_Data (чтение данных идентификации и обслуживания), 450
Инструкция ModuleStates (чтение информации о состоянии модуля), 474
Использование памяти, 1397
Светодиодная индикация, 1387
Таблица текущего контроля, 1404
Диагностика, сокращение событий безопасности, 114
Динамическая привязка, 207
Длина PtP-сообщения, 1134
Добавление входов или выходов в LAD и FBD инструкции, 43
Добавление нового устройства
CPU, 152
Идентификация имеющегося оборудования, 154
Не специфицированный CPU, 154
Документация, 4
Доступ
Пользовательские веб-страницы, 1092
Доступные устройства
Форматирование карты памяти, 1396
Доступные устройства, обновление прошивки, 1395

З

Загружаемая память, 28
CPU 1211C, 1444
CPU 1212C, 1457
CPU 1214C, 1470
CPU 1215C, 1484
CPU 1217C, 1499
Пользовательские веб-страницы, 1092
Загрузка
Обновление прошивки, 147
Сертификат безопасности Siemens на PC, 1042
Загрузка в CPU
DB для пользовательских веб-страниц, 1091
Программа пользователя, 209
Проект, 209
Просмотр MAC- и IP-адресов, 824

Загрузка в рабочем состоянии RUN
Глобальные установки для резерва памяти, 1418
Загрузка без повторной инициализации, 1416
Загрузка выбранных блоков, 1414
Загрузка не удалась, 1419
из STEP 7, 1413
Критерии безопасности, 1419
Необходимые условия, 1412
Обзор, 1410
Ограничения, 1418
Ошибка компиляции, 1415
Расширенный интерфейс блока, 1416
Резерв памяти и постоянный резерв памяти, 1416
Задержка, 103
Задержка включения (TON), 227
Задержка включения/выключения RTS, 1127
Задержка выключения (TOF), 227
Заказные номера
CPU, 1637
CPU 1214FC, CPU 1215FC, 1640
CSM 1277 Ethernet коммутатор, 1641
FS сигнальные модули, 1640
STEP 7, 1653
WinCC, 1653
Базовые панели HMI, 1641
Блок питания PM 1207, 1641
Имитаторы, 1642
Карты памяти, 1641
Коммуникационные интерфейсы (CM, CB und CP), 1639
Концевой фиксатор, 1642
ПО для программирования, 1653
Программное обеспечение визуализации, 1653
Расширительный кабель, 1642
Сигнальные модули, 1637
Сигнальные платы (SB) и батарейные платы (BB), 1638
Соединители и клеммные соединения, 1640
Терминальные блоки, 1642
Замена CPU V3.0 на CPU V4.x, 1655
Замена модулей, 50
Замена устройства
Замена CPU V3.0 на CPU V4.x, 1655
Порядок действий, 166
Запись в DB, I/O или память, 199, 271
Запрещенные TSAP и номера портов, 931
Запуск
Значения в таблице текущего контроля, 1406
Трассировка, 1421
Захват импульсов, 170, 174
Защита
CPU защита от несанкционированного доступа, 203
Защита ноу-хау для блока кода, 206
Защита ноу-хау
Защита паролем, 206
Защита ноу-хау, отображение с веб-сервером, 1043
Защита от копирования
Привязка к CPU, карте памяти или паролю, 207
Защита от неправильной полярности, 1440
Защита от несанкционированного доступа, CPU, 203
Защита паролем
ENDIS_PW (ограничение и восстановление действия пароля), 316
Блок кода, 206
Доступ к CPU, 203
Защита от копирования, 207
Привязка к CPU, карте памяти или паролю, 207
Пустая трансферная карта, 149
Утерянный пароль, 149

И

И (логическая операция), 334
Идентификация CPU, отображение с веб-сервером, 1043
Идентификация для выгрузки онлайн-CPU, 154
Избранное, панель инструментов, 41
Изменение настроек STEP 7, 45
Изменить
Zustand im Programmiereditor, 1402
Переменные через веб-сервер, 1055
Таблица текущего контроля, 1404
Изменить устройство, 166
Измерения, операции трассировки, 1422
ИЛИ (логическая операция), 334
Имитаторы, 1625
Имитаторы входных сигналов, 1625
Импульс (TP), 227
Импульсные выходы, 492
Индекс из переменных для массива, 278
Индексация массива с помощью переменных, 278
Индикатор первого цикла, 112
Индуктивные нагрузки, 79
Инструкции
&-блок (логическая операция AND в FBD), 220
-( )- (NO-катушка), 221
-(/)- (NC-катушка), 221

-(N)- (установка операнда при отрицательном фронте сигнала), 225
-(P)- (установка операнда при положительном фронте сигнала), 225
-(RESET_BF) (сброс битового поля), 223
-(SET_BF) (установка битового поля), 223
/=-блок (отрицание назначения в FBD), 221
-|/|- (NC-контакт), 219, 279, 283, 288, 293
-||- (NO-контакт), 219
-|N|- (опрос операнда на отрицательный фронт сигнала), 224
-|P|- (опрос операнда на положительный фронт сигнала), 224
=-блок (назначение FBD), 221
>=1-блок (логическая операция OR в FBD), 220
ABS (создание абсолютного значения), 251
ACOS (определение значения арккосинуса), 254
ACT_TINT (активация прерывания по времени), 430
ADD (сложение), 248
ASIN (определение значения арксинуса), 254
ATAN (определение значения арктангенса), 254
ATH (преобразование строки символов ASCII в шестнадцатеричное число), 364
ATTACH (назначить OB прерывающему событию), 421
ATTR_DB (чтение атрибутов блока данных), 544
CALCULATE, 41
CALCULATE (вычисление), 247
CAN_DINT (отмена прерывания по задержке времени), 432
CAN_TINT (удаление прерывания по времени), 430
CASE (SCL), 327
CEIL (округление числа с плавающей запятой до следующего целого числа в сторону увеличения), 304
Chars_TO_Strg (преобразование массива из CHAR в строку символов), 362
CONCAT (объединение строк символов), 368
CONTINUE (SCL), 331
CONV (преобразование значения), 299
COS (определение значения косинуса), 254
CountOfElements (запрос количества элементов массива ARRAY), 276
CREATE_DB (создание блока данных), 537
CTD (обратный счёт), 236
CTRL_HSC (управление высокоскоростным счетчиком), 587
CTRL_HSC_EXT (управление высокоскоростным счетчиком (расширенное)), 558
CTRL_PTO (вывод последовательности импульсов), 489
CTRL_PWM (широтно-импульсная модуляция), 487
CTU (прямой счёт), 236
CTUD (прямой и обратный счёт), 236
DataLogClose (закрытие архива данных), 522
DataLogCreate (создание архива данных), 512
DataLogNewFile (архив данных в новом файле), 524
DataLogOpen (открыть архив данных), 516
DataLogWrite (запись архива данных), 518
DB_ANY_TO_VARIANT (преобразование DB_Any в VARIANT), 309
DEC (уменьшение на 1), 251
DECO (декодирование), 335
DELETE (удаление символов из строки символов), 370
DELETE_DB (удаление блока данных), 546
DEMUX (демультиплексирование), 338
DETACH (отменить назначение OB прерывающему событию), 421
DIV (деление), 248
DPNRM_DG (чтение диагностических данных ведомого устройства DP), 417
DPRD_DAT (чтение непротиворечивых данных стандартного ведомого устройства DP), 409
DPWR_DAT (запись непротиворечивых данных стандартного ведомого устройства DP), 409
EN_AIRT (активация выполнения прерываний с более высоким приоритетом и асинхронных ошибок), 435
ENCO (кодирование), 335
ENDIS_PW (ограничение и восстановление действия пароля), 316
EQ_ElemType (сравнить тип данных элемента МАССИВА на предмет НЕСОВПАДЕНИЯ с типом данных переменной (тега)), 245
EQ_ElemType (сравнить тип данных элемента МАССИВА на предмет СОВПАДЕНИЯ с типом данных переменной (тега)), 245
EQ_Type (сравнить тип данных на предмет СОВПАДЕНИЯ с типом данных переменной (тега)), 245
EXIT (SCL), 332
EXP (определение экспоненциального значения), 254
EXPT (возведение в степень), 254
F_TRIG (установка переменной (тега) по отрицательному фронту сигнала), 226

FieldRead (чтение поля), 277
FieldWrite (запись поля), 277
FILL_BLK (заполнить блок), 265
FIND (поиск символов в строке символов), 373
FLOOR (округление числа с плавающей запятой до следующего целого числа в сторону уменьшения), 304
FOR (SCL), 329
FRAC (определение количества разрядов после запятой), 254
Gen_UsrMsg (создание диагностических сообщений пользователя), 436
GET (считывание данных из удаленного CPU), 1002
GET_DIAG (чтение информации системы диагностики), 479
GET_ERROR (локальное получение ошибки), 320
GET_ERROR_ID (локальное получение ID ошибки), 321
GetBlockName (считывание имени блока), 384
GetInstanceName (считывание имени экземпляра блока), 380
GetSymbolName (считывание имени переменной во входном параметре), 374
GetSymbolPath (запрос составного глобального имени входного параметра), 377
GetSymbolPath (запрос составного глобального имени экземпляра блока), 382
GOTO (SCL), 333
HTA (преобразование шестнадцатеричного числа в строку символов ASCII), 364
IF-THEN (SCL), 326
IN_Range (значение в диапазоне), 243
INC (увеличение на 1), 251
INSERT (вставка символов в строку символов), 371
INV (создание дополнения до единицы), 335
IO2MOD (определение аппаратного идентификатора по адресу ввода/вывода), 551
IS_ARRAY (проверка на МАССИВ), 246
IS_NULL (запрос на РАВЕНСТВО указателя НУЛЮ), 246
JMP (переход при RLO = 1), 311
JMP_LIST (определение списка переходов), 312
JMPN (переход при RLO = 0), 311
Label (метка перехода), 311
LED (чтение состояния светодиода), 449
LEFT (считывание левых символов в строке символов), 369
LEN (определение длины строки символов), 367
LIMIT (установка предельного значения), 253
LN (определение натурального логарифма), 254
LOG2GEO (определение слота по аппаратному идентификатору), 549
LOWER_BOUND (считывание нижней границы массива ARRAY), 268
MAX (вычислить максимум), 252
MAX_LEN (максимальная длина строки символов), 367
MB_CLIENT, 1200
MB_RED_CLIENT, 1221
MB_RED_SERVER, 1240
MC_CommandTable, 756
MC_MoveAbsolute (абсолютное позиционирование оси), 746
MID (считывание средних символов в строке символов), 369
MIN (вычислить минимум), 252
MOD (остаток от деления), 249
Modbus_Master (обмен данными через SIPLUS I/O или PtP-порт в качестве ведущего устройства Modbus RTU), 1274
Modbus_Slave (обмен данными через SIPLUS I/O или PtP-порт в качестве ведомого устройства Modbus RTU), 1282
MOVE (копирование значения), 256
MOVE_BLK (копирование области), 256
MUL (умножение), 248
MUX (мультиплексирование), 337
N (опрос операнда на отрицательный фронт сигнала), 224
N_TRIG (опрос RLO на отрицательный фронт сигнала), 225
N=-блок и N-катушка (установка операнда при отрицательном фронте сигнала), 225
NE_Type (сравнить тип данных на предмет НЕСОВПАДЕНИЯ с типом данных переменной (тега)), 245
NEG (создать двоичное дополнение), 250
NORM_X (нормализация), 305
NOT (НЕ) (инверсия RLO), 221
NOT_NULL (запрос на ОТЛИЧИЕ указателя от НУЛЯ), 246
NOT_OK (проверить на недостоверность), 244
OK (проверить на достоверность), 244
OUT_Range (значение вне диапазона), 243
P (опрос операнда на положительный фронт сигнала), 224
P_TRIG (опрос RLO на положительный фронт сигнала), 225

P=-блок и P-катушка (установка операнда при положительном фронте сигнала), 225
P3964_Config (конфигурирование протокола 3964(R)), 1152
PID_Compact (универсальный ПИД-регулятор со встроенной настройкой), 594
PID_Temp (универсальный ПИД-регулятор для регулирования температуры), 613
Port_Config (конфигурирование порта), 1141
PRVREC (предоставление набора данных), 415
PUT (запись данных в удаленный CPU), 1002
QRY_CINT (запрос параметров циклического прерывания), 426
QRY_DINT (запрос состояния прерывания по задержке времени), 432
QRY_TINT (запрос состояния прерывания по времени), 431
R (сброс выхода), 222
R_TRIG (установка переменной (тега) по положительному фронту сигнала), 226
RCVREC (получение набора данных), 412
RD_ADDR (определение адреса ввода/вывода по аппаратному идентификатору), 552
RD_LOC_T (считывание местного времени), 346
RD_SYS_T (считывание реального времени), 346
RDREC (чтение набора данных), 387
RE_TRIGR, 107
RE_TRIGR (перезапуск времени контроля цикла), 318
READ_BIG (чтение данных в формате Big-Endian), 272
READ_DBL (чтение из блока данных в загружаемой памяти), 541
READ_LITTLE (чтение данных в формате Little-Endian), 272
Receive_Config (конфигурация приема), 1147
Receive_P2P (прием данных по коммуникации "точка-точка"), 1158
RecipeExport (экспорт рецептуры), 504
RecipeImport (импорт рецептур), 506
REPEAT (SCL), 331
REPLACE (замена символов в строке символов), 372
RESET_BF (сброс битового поля), 223
RET (возвращаемое значение), 315
RETURN (SCL), 333
RIGHT (считывание правых символов в строке символов), 369
ROL (циклический сдвиг влево) и ROR (циклический сдвиг вправо), 341
ROUND (округление числа), 303
RS (триггер сброса/установки), 223
RT (сброс таймера), 227
RTM (счетчик часов работы), 350
RUNTIME (измерение времени выполнения программы), 323
S (установка выхода), 222
S_CONV (преобразование строки символов), 353
S_MOV (перемещение строки символов), 353
SCALE_X (масштабирование), 305
SEL (выбор), 337
Send_Config (конфигурация передачи), 1144
Send_P2P (передача данных в рамках коммуникации "точка-точка"), 1154
SET_BF (установка битового поля), 223
SET_CINT (параметрирование циклического прерывания), 424
Set_Features (установка расширенных функций), 1164
SET_TIMEZONE (установка часового пояса), 349
SET_TINTL (установка даты и времени для прерывания по времени), 428
SGN_GET (получить сигналы RS232), 1161
SHL (сдвиг влево) и SHR (сдвиг вправо), 340
Signal_Set (установка сигналов RS232), 1162
SIN (определение значения синуса), 254
SQR (определение квадрата), 254
SQRT (определение квадратного корня), 254
SR (триггер сброса/установки), 223
SRT_DINT (запуск прерывания по задержке времени), 432
STP (завершение программы), 319
Strg_TO_Chars (преобразование строки символов в массив из CHAR), 362
STRG_VAL (преобразование строки символов в числовое значение), 353
SUB (вычитание), 248
SWAP (поменять месторасположение), 267
SWITCH (распределение переходов), 313
T_ADD (суммирование времени), 344
T_COMBINE (объединение времени), 345
T_CONFIG (настройка интерфейса), 919
T_CONV (преобразование и извлечение значений времени), 343
T_DIAG, 887
T_DIFF (разность времен), 345
T_RESET, 885
T_SUB (вычитание времени), 344
TAN (определение значения тангенса), 254
TCON, 864

TDISCON, 864
TM_Mail (передача электронного сообщения), 1379
TOF (задержка выключения), 227
TON (задержка включения), 227
TONR (задержка включения с запоминанием), 227
TP (импульс), 227
TRCV, 864
TRCV_C, 843, 938
TRUNC (округление до целого числа), 303
TSEND, 864
TSEND_C, 843, 937
TURCV (получение данных через Ethernet (UDP)), 914
TUSEND (передача данных через Ethernet (UDP)), 914
UFILL_BLK (заполнить блок без прерываний), 265
UMOVE_BLK (непрерывное копирование области), 256
UPPER_BOUND (считывание верхней границы массива ARRAY), 269
USS_Port_Scan (обработка коммуникации через сеть USS), 1182
USS_Read_Param (считывание параметров из привода), 1187
VAL_STRG (преобразование числового значения в строку символов), 353
VARIANT_TO_DB_ANY (преобразование из VARIANT в DB_ANY), 308
VariantGet (чтение значения переменной (тега) VARIANT), 274
VariantPut (запись значения переменной (тега) VARIANT), 275
WHILE (SCL), 330
WR_LOC_T (установка местного времени), 346
WR_SYS_T (установка реального времени), 346
WRIT_DBL (запись в блок данных в загружаемой памяти), 541
WRITE_BIG (запись данных в формате Big-Endian), 272
WRITE_LITTLE (запись данных в формате Little-Endian), 272
WRREC (запись набора данных), 387
WWW (синхронизация пользовательских веб-страниц), 1089
XOR (ИСКЛЮЧАЮЩЕЕ ИЛИ) (логическая операция), 334
x-блок (логическая операция XOR в FBD), 220
Варианты PEEK и POKE, 199, 271
Версии инструкций, 44, 843, 856, 863, 875, 1179, 1199, 1269, 1315, 1328, 1346
Время, 343
Вставить, 40
Дата, 343
Десериализация, 260
Добавление входов или выходов в LAD и FBD инструкции, 43
И (логическая операция), 334
Избранное, 41
ИЛИ (логическая операция), 334
Инструкции преобразования в SCL, 300
Инструкция (опрос расширенных функций), 1163
Инструкция DeviceStates (считывание состояния для модуля системы ввода/вывода), 468
Инструкция GEO2LOG (определение аппаратного идентификатора через слот), 548
Инструкция Get_IM_Data (чтение данных идентификации и обслуживания), 450
Инструкция MC_ChangeDynamic (изменение настроек динамики для оси), 759
Инструкция MC_Halt (приостановка оси), 744
Инструкция MC_Home (выполнение движения оси к референтной точке), 741
Инструкция MC_MoveJog (перемещение оси в толчковом режиме), 754
Инструкция MC_MoveRelative (относительное позиционирование оси), 749
Инструкция MC_MoveVelocity (перемещение оси с заданной скоростью), 751
Инструкция MC_Power (деблокировка/блокировка оси), 737
Инструкция MC_ReadParam (чтение параметров технологического объекта), 763
Инструкция MC_Reset (подтвердить ошибку), 740
Инструкция MC_WriteParam (запись параметров технологического объекта), 761
Инструкция Modbus_Comm_Load (конфигурирование SIPLUS I/O или порта на PtP модуле для Modbus RTU), 1270
Инструкция ModuleStates (чтение информации о состоянии модуля), 474
Инструкция RALRM (получение прерывания), 395
Инструкция Receive_Reset (удалить данные в принимающем буфере), 1160
Инструкция USS_Drive_Control (обмен данными с приводом), 1184

Инструкция USS_Write_Param (изменить параметры в приводе), 1188
Календарь, 343
Коды состояния USS, 1190
Масштабировать аналоговые значение, 42
Мониторинг, 1402
Общие параметры, 930
Перетаскивание, 40
Перетаскивание между редакторами, 46
Распределенная периферия AS-i, 386
Распределенная периферия PROFIBUS, 386
Распределенная периферия PROFINET, 386
Расширяемые инструкции, 44
Сброс выхода, 222
Сериализация, 263
Состояние, 1402
Состояние или значение мониторинга, 1401
Сравнение значений, 242
Старая инструкция TRCV_C (получение данных через Ethernet (TCP)), 856
Старая инструкция TSEND_C (передача данных через Ethernet (TCP)), 856
Старые инструкции TCON, TDISCON, TSEND и TRCV, 876
Старые коды состояния USS, 1326
Столбцы и заголовки, 44, 843, 856, 863, 875, 1179, 1199, 1269, 1315, 1328, 1346
Таймер, 227
Управление перемещением, 735
Управления программой (SCL), 325
Установка выхода, 222
Функция принудительной установки, 1409
Часы, 346
Инструкции PROFINET
T_CONFIG (настройка интерфейса), 919
T_DIAG, 887
T_RESET, 885
TCON, 864
TDISCON, 864
TRCV, 864
TRCV_C, 843, 938
TSEND, 864
TSEND_C, 843
TURCV (получение данных через Ethernet (UDP)), 914
TUSEND (передача данных через Ethernet (UDP)), 914
Старая инструкция TRCV_C (получение данных через Ethernet (TCP)), 856
Старая инструкция TSEND_C (передача данных через Ethernet (TCP)), 856
Старые инструкции TCON, TDISCON, TSEND и TRCV, 876
Инструкции PtP, возвращаемые значения, 1139
Инструкции управления перемещением, 735
Инструкции, скорость выполнения, 1446, 1459, 1472, 1486, 1501
Инструкции, устаревшие
MB_CLIENT (обмен данными с использованием PROFINET в качестве клиента Modbus TCP), 1329
MB_COMM_LOAD (конфигурирование порта на PtP модуле для Modbus RTU), 1347
MB_MASTER (обмен данными через PtP-порт в качестве ведущего устройства Modbus), устаревшая, 1350
MB_SERVER (обмен данными с использованием PROFINET в качестве сервера Modbus TCP), 1336
MB_SLAVE (обмен данными через PtP-порт в качестве ведомого устройства Modbus), устаревшая, 1356
PORT_CFG (динамическое конфигурирование коммуникационных параметров), 1298
RCV_CFG (динамическое конфигурирование параметров последовательного приема), 1301
RCV_PTP (разрешить прием сообщений), 1309
RCV_RST (удалить данные в принимающем буфере), 1311
SEND_CFG (динамическое конфигурирование параметров последовательной передачи), 1300
SEND_PTP (отправить данные из передающего буфера), 1307
SGN_GET (запросить сигналы RS232), 1312
SGN_SET (определить сигналы RS232), 1313
USS_DRV (обмен данными с приводом), 1320
USS_PORT (обработка коммуникации через сеть USS), 1318
USS_RPM (считывание параметров из привода), 1323
USS_WPM (изменение параметров в приводе), 1324
Инструкция (опрос расширенных функций), 1163
Инструкция DeviceStates (считывание состояния для модуля системы ввода/вывода), 468
Инструкция GEO2LOG (определение аппаратного идентификатора через слот), 548
Инструкция Get_IM_Data (чтение данных идентификации и обслуживания), 450

Инструкция MC_ChangeDynamic (изменение настроек динамики для оси), 759
Инструкция MC_Halt (приостановка оси), 744
Инструкция MC_Home (выполнение движения оси к референтной точке), 741
Инструкция MC_MoveJog (перемещение оси в толчковом режиме), 754
Инструкция MC_MoveRelative (относительное позиционирование оси), 749
Инструкция MC_MoveVelocity (перемещение оси с заданной скоростью), 751
Инструкция MC_Power (деблокировка/блокировка оси), 737
Инструкция MC_ReadParam (чтение параметров технологического объекта), 763
Инструкция MC_Reset (подтвердить ошибку), 740
Инструкция MC_WriteParam (запись параметров технологического объекта), 761
Инструкция Modbus_Comm_Load (конфигурирование SIPLUS I/O или порта на PtP модуле для Modbus RTU), 1270
Инструкция RALRM (получение прерывания), 395
Инструкция Receive_Reset (удалить данные в принимающем буфере), 1160
Интеллектуальное IO устройство (Intelligent IO Device)
  Функциональность, 943
Интервал между символами, 1133
Интерфейс пользователя
  STEP 7 проектное и портальное представление, 39
Источники информации, 4

К

Кабель
  Расширение, 1628
  Сетевая коммуникация, 1118
Календарь, 343
Карта памяти, 1624
  Вставить в CPU, 137
  Конфигурирование параметров запуска, 139
  Обзор, 136
  Обновление прошивки, 147
  Ошибка из-за несовместимости, 1388
  Программная карта, 144
  Пустая трансферная карта для утерянного пароля, 149
  Трансферная карта, 140
  Утерянный пароль, 149
Карта памяти CPU
  Вставить, 137
  Программная карта, 144
  Трансферная карта, 140
Карты задач
  Столбцы и заголовки, 1179, 1199, 1269, 1315, 1328, 1346
Каскадирование ПИД-регуляторов, 620
Каталоги, языки для пользовательских веб-страниц, 1105
Категория перенапряжения, 1440
Катушки, 219
Классы ошибок PtP, 1141, 1297
Клиентская поддержка, 3
Кольцевая топология, 971
Кольцевой порт, 978
Коммуникации точка-точка, 1120
Коммуникационная плата (CB)
  CB 1241 RS485, 1620
  RS485, 1117
  Демонтаж, 62
  Добавление модулей, 156
  Конфигурация параметров, 174
  Конфигурация устройства, 151
  Монтаж, 62
  Обзор, 31
  Программирование, 1165
  Светодиодная индикация, 1117, 1387
  Сравнительная таблица, 31
Коммуникационные интерфейсы
  CB 1241 RS485, 1620
  CM 1241 RS232, 1621
  RS232 и RS485, 1117
  Добавление модулей, 156
  Конфигурация, 1123
  Конфигурация устройства, 151
  Программирование, 1165
  Светодиодная индикация, 1387
  Сравнительная таблица модулей, 31
Коммуникационные интерфейсы, 3964(R), 1136
Коммуникационные модули RS232 и RS485, 1117
Коммуникационный модуль (CM)
  CM 1241 RS232, 1621
  CM 1241 RS422/RS485, 1622
  RS232 и RS485, 1117
  Демонтаж, 66
  Добавление CM 1243-5 (ведущее устройство DP), 987
  Добавление модулей, 156
  Добавление модуля ведущего AS-i устройства CM1243-2, 992
  Конфигурация параметров, 174
  Конфигурация устройства, 151

Конфигурирование PtP программы-примера, 1168
Монтаж, 66
Обзор, 32
Потребляемая мощность, 1633
Прием данных, 1158, 1309
Программирование, 1165
Светодиодная индикация, 1117, 1387
Сравнительная таблица, 31
Коммуникационный процессор (CP)
Добавление модулей, 156
Конфигурация параметров, 174
Конфигурация устройства, 151
Обзор, 32
Сравнительная таблица, 31
Коммуникация
IP-адрес, 819
MAC-адрес, 819
PROFINET и PROFIBUS, 805
TCON_Param, 835
Адрес AS-i, 994
Адрес PROFIBUS, 988
Активная/пассивная, 812, 835, 1008
Аппаратное подключение, 932
Архитектура опроса, 1166
Время цикла, 108
Идентификаторы соединений, 832
Количество соединений (PROFINET/ PROFIBUS), 808
Коммуникационная нагрузка, 108
Конфигурация, 812, 835, 1008
Параметры передачи и приема, 1126
Потеря, удаление или вставка модулей, 98
Протоколы, 830
Свойство синхронизации времени (PROFINET), 826
Сетевое соединение, 811
Сеть, 931
Управление потоком, 1125
коммуникация S7
Конфигурирование соединения, 813
Коммуникация TCP/IP, 828
Коммуникация Teleservice
TM_Mail (передача электронного сообщения), 1379
Компактный коммутационный модуль CSM 1277, 1628
Компактный коммутационный модуль, CSM 1277, 1628
Компенсация холодного спая, термопара, 1554, 1599
Контактная информация, 3, 166
Контакты, 219
Максимальная нагрузка, 1618
Контроль времени цикла (инструкция RE_TRIGR), 318
Контроль и изменение переменных через веб-сервер, 1055
Контроль переменных через веб-сервер, 1055
Конфигурация
ASi, 994
Ethernet порт, 819
HSC (высокоскоростные счетчики), 574
IP-адрес, 819
MAC-адрес, 819
PROFIBUS, 988
PROFINET порт, 819
RS422, рабочие режимы, 1171
RS485, рабочие режимы, 1173
Адрес PROFIBUS, 988
Время цикла, 107
Добавление модулей, 156
Загрузка в CPU, 209
Идентификация, 154
Инструкции PID_Compact и PID_3Step, 626
Инструкции PID_Temp, 629
Интерфейсы, 1123
Коммуникационная нагрузка, 108
Коммуникационные интерфейсы, 1123
Коммуникация PLC/PLC, 935
Модули, 174
Параметры CPU, 167
Параметры запуска, 139
Порт AS-i, 993
Принять сообщение, 1128
Свойство синхронизации времени (PROFINET), 826
Сетевое соединение, 811
Конфигурация CPU
Импульсные каналы, 494
Коммуникация с HMI, 934
Контроль времени цикла, 107
Несколько CPU, 935
Рабочие параметры, 167
Свойства модуля, 174
Конфигурация оборудования, 151
ASi, 994
Ethernet порт, 819
PROFIBUS, 988
PROFINET, 819
Добавление модулей, 156
Добавление нового устройства, 152
Загрузка в CPU, 209
Идентификация, 154

Конфигурирование CPU, 167
Конфигурирование модулей, 174
Порт AS-i, 994
Сетевое соединение, 811
Конфигурация отправленного сообщения, 1127
Конфигурация параметров
LENGH и BUFFER для Send_P2P, 1157
Передача, 938
Прием, 939
Конфигурация передачи сообщения, 1127
PtP устройство, конфигурация, 1127
Пример программы PtP, 1168
Конфигурация порта, 1123
Error, 1143, 1299
Пример программы PtP, 1168
Конфигурация портов, 3964(R), 1136
Конфигурация приема сообщения
PtP устройство, конфигурация, 1128
Пример программы PtP, 1169
Конфигурация сообщения
Передача, 1127
Прием, 1128
Конфигурация устройства, 151, 932
ASi, 994
Ethernet порт, 819
PROFIBUS, 988
PROFINET, 819
Добавление модулей, 156
Добавление нового устройства, 152
Загрузка в CPU, 209
Идентификация, 154
Изменение типа устройства, 166
Конфигурирование CPU, 167
Конфигурирование модулей, 174
Не вставленные модули, 50
Порт AS-i, 994
Свойство синхронизации времени (PROFINET), 826
Сетевое соединение, 811
Конфигурация, 3964(R)
Интерфейсы, 1136
Коммуникационные интерфейсы, 1136
Приоритет и параметры протокола, 1137
Конфигурирование интерфейса
Инструкции, 1166
Конфигурирование параметров передачи, 812, 938, 1008
Конфигурирование параметров приема, 939
Конфигурирование пользователей, веб-сервер, 1028
Конфигурирование сообщений
Инструкции, 1166
Копирование блоков из Онлайн-CPU, 214
Копирование, вырезание и вставка в STEP 7, 50
Куки, siemens_automation_language, 1105

Л

Ламповые нагрузки, 78
Летнее время TimeTransformationRule, 348
Линейное программирование, 180
Линия на паузе, 1127
Лист ожидания, 103
Лицензии OPC UA, 1653
Лицензии, OPC UA, 1653
Логические значения или битовые значения, 117
Логический анализатор, 1421
Локальные данные
Использование блоками, 120
Максимум на каждый уровень приоритета OB, 120

М

Максимальная длина сообщения, 1133
Максимальное количество соединений с веб-сервером, 1112
Маркировка CE, 1433
Маска подсети, 819
Массивы, доступ к элементам, 278
Масштабирование аналоговых значений, 306
Масштабировать аналоговые значение, 42
Мгновенный снимок значений DB, 1403
Местное время
RD_LOC_T (считывание местного времени), 346
WR_LOC_T (установка местного времени), 346
Мобильное устройство, доступ к веб-серверу, 1031
Мобильные устройства
Структура веб-страницы, 1037
Модернизация CPU версии V3.0 до версии V4.x, 1655
Модули
Коммуникационные платы (CB), 31
Коммуникационный модуль (CM), 32
Коммуникационный процессор (CP), 32
Конфигурирование параметров, 174
Сигнальная плата (SB), 31
Сигнальные модули (SM), 32
Сравнительная таблица, 31
Тепловая зона, 54, 58

Модули CANopen
021620-B, 021630-B, 1629
Модуль потенциометров
Технические данные, 1627
Мониторинг
Время цикла, 1397
Использование памяти, 1397
Регистрация и сброс значений DB, 1403
Состояние LAD, 1402
Состояние LAD и использование таблицы текущего контроля, 1401
Таблица принудительно задаваемых значений, 1408
Таблица текущего контроля, 1404
Функция принудительной установки, 1409
Мониторинг программы, 215
Монтаж
CPU, 60
TS адаптер на DIN-рейке, 72
Заземление, 76
Индуктивные нагрузки, 79
Коммуникационная плата (CB), 62
Коммуникационный модуль (CM), 66
Ламповые нагрузки, 78
Монтаж TS адаптера на стену, 73
Монтажные размеры, 58
Необходимые условия, 37
Обзор, 53, 59
Охлаждение, 54
Поток воздуха, 54
Правила электромонтажа, 74, 76
Развязка, 75
Размеры, 58
Расстояние, 54
Расширительный кабель, 68
Руководства, 53
Руководство по изоляции, 75
Сигнальная плата (SB), 62
Сигнальные модули (SM), 32
Сигнальный модуль (SM), 64
Тепловая зона, 54, 58
Терминальный блок, 67
Морские сертификаты, 1436

H

Назначение типов перечисления, пользовательские веб-страницы, 1082
Накопитель времени (TONR), 227
Настройки, 45
Не вставленные модули, 50
Не специфицированный CPU, 154
Необходимые условия, PC, 37
Необходимые условия, установка, 37
Нерегулируемое управление перемещением
PTO, 666
Конфигурирование оси, 666
Несколько AWP определений переменной, 1085
Новые функции, 35
Номера портов
запрещенные, 931
Назначение участникам коммуникации, 828
Нормализация аналоговых значений, 306

O

Области памяти
Адресация логических значений или битовых значении, 117
Образ процесса, 116
Прямой доступ, 116
Области памяти, отображение с веб-сервером, 1043
Область памяти A
Импульсные выходы, 492
Конфигурирование импульсных каналов, 494
Область памяти I
Адреса периферийных входов (таблица принудительно задаваемых значений), 1408
Мониторинг, 1401
Мониторинг LAD, 1402
Принудительное присваивание значений, 1408
Таблица принудительно задаваемых значений, 1408
Таблица текущего контроля, 1401
Функция принудительной установки, 1409
Обмен данными между системами ввода-вывода, 947
Обновление пользовательских веб-страниц, 1072
Обновление прошивки
из STEP 7, 1394
с помощью карты памяти, 147
через веб-сервер, 1049
Обработка события и лист ожидания, 103
Образ процесса
Запись выходов с SETIO, 391
Мониторинг, 1402
Передача области образа процесса с SETIO_PART, 394
Принудительное присваивание значений, 1408
Состояние, 1402, 1408
Состояние или значение мониторинга, 1401

Считывание входов с помощью GETIO, 390
Функция принудительной установки, 1409
Чтение области образа процесса с GETIO_PART, 392
ОВ прерывания по ошибке времени, 95
ОВ циклических прерываний, 93
Ограничение рывка, 735
Ограничения
Веб-сервер, 1112
Пользовательские веб-страницы, 1092
Ограничения для куки, стандартные веб-страницы, 1113
Окна задач
Столбцы и заголовки, 44, 843, 856, 863, 875
Онлайн
IP-адрес, 1393
Буфер диагностики, 1397
Время суток, 1393
Время цикла, 1397
Выход в онлайн, 1390
Инструменты, 1401
Использование памяти, 1397
Назначение IP-адреса, 817
Панель оператора, 1397
Панель управления, 47
Принудительное присваивание значений, 1408
Регистрация и сброс значений DB, 1403
Состояние, 1402
Состояние или значение мониторинга, 1401
Сравнение и синхронизация, 1399
Таблица текущего контроля, 1401, 1402, 1404
Функция принудительной установки, 1409
Экранные кнопки RUN/STOP, 46
Онлайн имена устройств
PROFINET IO, 1391
Онлайн инструменты и диагностика
Загрузка в рабочем состоянии RUN, 1410
Операции по фронту, положительному и отрицательному, 224
Определение типов перечисления, пользовательские веб-страницы, 1081, 1082
Оптимизированные блоки данных, 187
Организационный блок
Вызов, 91
Вызов блоков кода в программе пользователя, 182
Защита ноу-хау, 206
Классы приоритетов, 91
Конфигурирование принципа работы, 184
Линейное и структурное программирование, 180
Несколько ОВ цикла, 184
Обзор, 83
Обработка, 183
Обработка при запуске, 90
Согласование с временной памятью, 120
Создание, 184
Функция, 91
Циклическое прерывание, 93
Организационный блок (ОВ)
Чтение начальной информации с RD_SINFO, 440
Открытые коммуникационные соединения пользователя
Создание соединения и передача данных с помощью TSEND_C, 843
Создание соединения и передача данных с помощью старой инструкции TSEND_C, 856
Создание соединения и считывание данных с помощью TRCV_C, 843
Создание соединения и считывание данных с помощью старой инструкции TRCV_C, 856
Отступ, вентиляция и охлаждения, 54
Охлаждение, 54
Ошибка
Диагностическая ошибка, 96
Ошибка по времени, 95
Часто встречающиеся в расширенных инструкциях ошибки, 555
Ошибка из-за неизвестной версии CPU, 1388
Ошибка из-за несовместимой версии CPU, 1388
Ошибка обработки сигнала, 1162, 1163, 1313, 1314
Ошибки выполнения программы при передаче, 1156, 1308
Ошибки конфигурации передачи, 1146, 1301
Ошибки конфигурации приема, 1151, 1306

П

Память
I (образ процесса входов), 118
L (память локальных данных), 116
M (битовая память, меркеры), 120
Q (образ процесса выходов), 119
Адреса периферийных входов (таблица принудительно задаваемых значений), 1408
Временная память, 120
Загружаемая память, 109
Мониторинг использования памяти, 1397
Рабочая память, 109
Системная память, 111

Сохраняющая память, 109
Тактовая память, 111
Панели (HMI), 32
Панель оператора, 32
Панель управления, 47
Папки, языки для пользовательских веб-страниц, 1105
Параметр BUFFER, Send_P2P, 1157
Параметр LENGTH, Send_P2P, 1157
Параметрирование, 185
Параметры запуска, 139
Пассивная/активная коммуникация
Конфигурирование участников, 1008
Пауза, 1127, 1128
Передача данных, запустить, 1154, 1307
Переключение языков, пользовательские веб-страницы, 1105
Перекрестная ссылка для отображения использования, 217
Переменная (тег)
Наложение, 134
Состояние или значение мониторинга, 1401
Срез, 133
Функция принудительной установки, 1409
Перетаскивание между редакторами, 46
Переход от режима RUN в STOP, 115
ПИД
Algorithmus PID_3Step, 591
Algorithmus PID_Compact, 591
PID_3STEP (ПИД-регулятор с настройкой для вентилей), 602
PID_Compact (универсальный ПИД-регулятор со встроенной настройкой), 594
PID_Temp (универсальный ПИД-регулятор для регулирования температуры), 613
Каскадирование регуляторов, 620
Обзор, 591
Параметр ErrorBit из PID_3Step, 610
Параметр ErrorBit из PID_Compact, 599
Параметр ErrorBit из PID_Temp, 624
Предельные значения для технологического значения у регулятора PID_Compact, 597
ПИД настройка вентилей, 602
Планшет, доступ к веб-серверу, 1031
Поддерживаемые для веб-сервера браузеры, 1025
Поддержка, 3
Подключения
Ethernet протоколы, 1006
S7-соединение, 1006
Веб-сервер, 1112
Идентификаторы соединений, 832
Конфигурация, 835
Типы коммуникации, 805
Типы, соединения с несколькими узлами, 1006
Участник, 812, 1008
Позиция символа, длина сообщения, 1134
Пользовательские веб-страницы, 1024, 1071
AWP-команды для доступа к данным S7-1200, 1072
HTML-список, 1099
Активация и деактивация через управляющий DB, 1108
Активация с помощью WWW инструкции, 1089
Генерация программных блоков, 1088
Доступ через PC, 1092
Загрузка соответствующих DB в CPU, 1091
Запись переменных, 1075
Запись специальных переменных, 1079
Импорт фрагментов, 1084
Конфигурация STEP 7, 1087
Конфигурация для нескольких языков, 1108
Конфигурирование, 1087
несколько языков, 1105
Обновление, 1072
Обработка специальных символов, 1085
Ограничения для загружаемой памяти, 1092
Пример, 1094
Программирование в STEP 7, 1089
Ручное управление фрагментным DB, 1108
Создание с помощью HTML редактора, 1071
Создание фрагментов, 1083
Удаление программных блоков, 1089
Установка нескольких языков, 1108
Чтение переменных, 1074
Чтение специальных переменных, 1077
Портальное представление, 39
Последовательная коммуникация, 1120
Последовательность движений (MC_CommandTable), 756
Последовательность символов
Конец сообщения, 1133
Начало сообщения, 1129
Потеря связи с CPU на модулях, 98
Поток воздуха, 54
Потребляемый ток, 55, 1633
Правила для кавычек, веб-сервер, 1085
Правила электромонтажа, 76
Заземление, 76
Необходимые условия, 74
Отступ для вентиляции и охлаждения, 54
Преобразование (инструкции SCL), 300

Прерывания
ATTACH (назначить OB прерывающему событию), 421
CAN_DINT (отмена прерывания по задержке времени), 432
DETACH (отменить назначение OB прерывающему событию), 421
QRY_DINT (запрос состояния прерывания по задержке времени), 432
SRT_DINT (запуск прерывания по задержке времени), 432
Задержка прерывания, 103
Обзор, 91
Прерывания по задержке времени, 432
Привод MicroMaster, подключение, 1192
Приводы, настройка привода MM4, 1194
Привязка к CPU, карте памяти или паролю, 207
Пример
Выбор уровня для активного переключателя референтной точки, 733
Выбор уровня для пассивного переключателя референтной точки, 732
Добавление преобразователя частоты SINAMICS S120, 688
Конфигурирование обнаружения фронта на предельной позиции или входе переключателя референтной точки, 720
Примеры управления перемещением
Velocity (Скорость), 770
Поведение в толчковом режиме, 774
Примеры, Modbus
MB_CLIENT для нескольких запросов с общим TCP соединением, 1263
MB_CLIENT для нескольких запросов с различными TCP соединениями, 1264
MB_SERVER для нескольких соединений TCP, 1262
Modbus TCP MB_CLIENT для координации нескольких запросов, 1266
Modbus TCP MB_CLIENT запрос записи для образа процесса выходов, 1265
Modbus TCP, MB_CLIENT-параметры соединения, 1205
Modbus TCP, MB_SERVER-параметры соединения, 1210
Modbus TCP, адрес регистра хранения, 1216
MODBUS TCP, примеры для параметра MB_HOLD_REG, 1213
Пример программы для ведомого устройства Modbus, 1296
Пример программы для ведущего устройства Modbus RTU, 1293
Примеры, Modbus CP в старых системах
Параметр MB_HOLD_REG, 1338
Примеры, Modbus RTU в старых системах
Программа ведомого устройства, 1365
Программа для ведущего устройства, 1363
Примеры, Modbus TCP в старых системах
MB_CLIENT для координации нескольких запросов Modbus TCP, 1345
MB_CLIENT для нескольких запросов с различными TCP соединениями, 1343
MB_CLIENT запрос записи для образа процесса выходов, 1344
MB_CLIENT: несколько запросов с общим Modbus TCP соединением, 1342
MB_SERVER для нескольких соединений TCP, 1341
Адресация регистра хранения, 1340
Примеры, Modbus в старых системах
Modbus RTU, адресация регистра хранения в старых системах, 1360
Примеры для Modbus RTU, параметр MB_HOLD_REG в старых системах, 1358
Примеры, PtP коммуникация
PtP коммуникация в старых системах, RCV_CFG, 1304
Receive_Config, 1149
Выполнение примера для эмулятора терминала, 1176
Длина сообщения в сообщении, 1134
Конфигурация, 1168
Программирование STEP 7, 1174
Условия для конца сообщения, 1134
Условия для начала сообщения, 1130
Эмулятор терминала, 1167, 1175
Примеры, веб-сервер
Доступ через мобильное устройство, 1032
Запись переменных, 1076, 1098
Запись специальных переменных, 1079, 1099
Объединение определений AWP, 1085
пользовательская веб-страница, 1094, 1099
пользовательская веб-страница для настройки языка, 1106
Программа STEP 7 для проверки на фрагменты, 1111
Псевдонимы, 1074, 1080
Специальные символы в AWP командах, 1086
Типы перечисления, 1081, 1082, 1097
Фрагментный DB, 1085
Чтение переменных, 1074, 1096
Чтение специальных переменных, 1078

Примеры, инструкции
ATH (преобразование строки символов ASCII в шестнадцатеричное число), 365
CALCULATE, 41
CONTINUE, SCL, 332
CTRL_PWM, 496
DECO (декодирование), 336
DeviceStates, PROFIBUS и PROFINET, 469
EXIT, SCL, 332
GET_DIAG и режимы, 485
GOTO (SCL), 333
HTA (преобразование шестнадцатеричного числа в строку символов ASCII), 366
LIMIT (установка предельного значения), 254
ModuleStates, PROFIBUS и PROFINET, 475
RETURN, SCL, 333
ROR (циклический сдвиг вправо), SCL, 341
RUNTIME (измерение времени выполнения программы), 324
S_CONV (преобразование строки символов), 360
SET_CINT выполнение циклического прерывания и параметры времени, 425
SHL (сдвиг влево), SCL, 340
STRG_VAL (преобразование строки символов в числовое значение), 361
SWAP (поменять месторасположение), 267
TM_MAIL, 1384
VAL_STRG (преобразование числового значения в строку символов), 362
Варианты PEEK и POKE, 199, 271
Десериализация, 261
Катушки таймера, 229
Сериализация, 264
Примеры, коммуникация
I-устройство в качестве IO устройства и IO контроллера, 947
I-устройство общего доступа, 961
T_CONFIG, изменение IP параметров der NTP-сервера, 929
T_CONFIG, изменение IP-параметров и имен устройств PROFINET IO, 928
T_CONFIG, изменить IP-параметр, 926
Адресация ведомого устройства ASi, 994
Коммуникация между CPU через соединения TSEND_C и TRCV_C, 834
Коммуникация между CPU, которые используют два отдельных соединения для отправки и получения данных., 832
Коммуникация между CPU, которые используют одно общее соединение для отправки и получения данных., 833
Конфигурирование S7-соединение PROFINET, 1012
Конфигурирование S7-соединения PROFIBUS, 1014
Протокол коммуникации PROFINET, 828
Телеуправление, 1373
Устройство общего доступа, 956
Примеры, коммуникация USS
Оповещение об ошибках USS коммуникации, 1191
Оповещение об ошибках USS коммуникации в старых системах, 1327
Примеры, ПИД
PID_3Step, настройки конфигурации, 628
PID_Compact, настройки конфигурации, 627
PID_Temp, настройки конфигурации, 630
Примеры, разные
AT наложение для переменной (тега), 135
CPU 1217C дифференциальный вход и использование, 1514
CPU 1217C дифференциальный выход и использование, 1516
вложенные операторы CASE, SCL, 328
Доступ к элементам массива, 278
Загрузка выбранных блоков в рабочем состоянии RUN, 1414
Обработка ENO в SCL, 202
Обработка аналоговых значений, 122, 306
Перетаскивание между редакторами, 46
Подключение S7-1200 IO-Link-Master, 1569
Программа архивации данных, 532
Расчет баланса мощностей, 1635
Рецептура, 500, 508
Срез тегированного типа данных, 134
Управление конфигурацией (управление опциями), 163
Функция трассировки и логического анализатора, 1423
Примеры, строковые инструкции при выполнении
GetBlockName, 385
GetInstanceName, 381
GetInstancePath, 382
GetSymbolName, 376
GetSymbolPath, 379
Примеры, управление перемещением
Конфигурации скорости импульсного выхода CPU 1211C, CPU 1212C, CPU 1214C и CPU 1215C, 660
Конфигурации скорости импульсного выхода CPU 1217C, 659
Конфигурирование MC_Tabelle с технологическим объектом, 710

Ограничение рывка, 735
Поведение оси, 766
Управление осью с помощью технологического модуля (TM) Pulse, 705
Эпюра скоростей для движения к референтной точке при управлении перемещением, 733
Принудительное присваивание значений, 1408
Входы и выходы, 1409
Входы периферии, 1408, 1409
Область памяти I, 1408, 1409
Таблица текущего контроля, 1404
Цикл, 1409
Приоритет
Классы приоритетов, 91
Приоритет при обработке, 103
Проверка непротиворечивости, 217
Проверка соединения, 887
Программа
Вызов блоков кода в программе пользователя, 182
Загрузка в CPU, 209
Защита паролем, 206
Карта памяти, 136
Классы приоритетов, 91
Копирование блоков из Онлайн-CPU, 214
Линейные и структурированные программы, 180
Организационные блоки (OB), 183
Привязка к CPU, карте памяти или паролю, 207
Программа для работы с рецептурами, пример, 508
Программа пользователя
Вставка инструкций, 40
Вызов блоков кода в программе пользователя, 182
Добавление входов или выходов в LAD и FBD инструкции, 43
Загрузка в CPU, 209
Защита паролем, 206
Избранное, 41
Карта памяти, 136
Копирование блоков из Онлайн-CPU, 214
Линейные и структурированные программы, 180
Организационные блоки (OB), 183
Перетаскивание между редакторами, 46
Привязка к CPU, карте памяти или паролю, 207
Программная карта, 136
Расширяемые инструкции, 44
Трансферная карта, 136
Программирование
Algorithmus PID_3Step, 591
Algorithmus PID_Compact, 591
FBD (функциональная блок-схема), 193
LAD (релейно–контактная схема), 192
PID_3STEP (ПИД-регулятор с настройкой для вентилей), 602
PID_Compact (универсальный ПИД-регулятор со встроенной настройкой), 594
PID_Temp (универсальный ПИД-регулятор для регулирования температуры), 613
RTM (счетчик часов работы), 350
SCL (Structured Control Language), 194, 196
Блок данных (DB), 83
Блок данных экземпляра (DB), 185
Вставка инструкций, 40
Вызов блоков кода в программе пользователя, 182
Вызовы блоков, 83
Добавление входов или выходов в LAD и FBD инструкции, 43
Избранное, 41
Инструкции PtP, 1165
Классы приоритетов, 91
Количество блоков кода, 83
Линейная программа, 180
Начальное значение FB, 185
Не вставленные модули, 50
Не специфицированный CPU, 154
Обзор ПИД-регулятора, 591
Перетаскивание между редакторами, 46
Поток сигналов (EN и ENO), 201
Правильные номера FC, FB и DB, 83
Привязка к CPU, карте памяти или паролю, 207
Рабочие состояния CPU, 87
Расширяемые инструкции, 44
Системное время, 346
Сравнение и синхронизация блоков кода, 1399
Структурированная программа, 180
Функциональный блок (FB), 83, 185
Функция (FC), 184
Программирование "точка-точка", 1165
Программирование STEP 7
Пользовательские веб-страницы, 1089
Пример программы PtP, 1174
Программирование переключателя языка для пользовательских веб-страниц, 1105
Программная карта
Вставить в CPU, 137
Конфигурирование параметров запуска, 139
Обзор, 136
Создание, 144
Программная структура, 182

Программное управление потоком, 1126
Проект
Загрузка в CPU, 209
Защита блока кода, 206
Парольная защита от несанкционированного доступа, 203
Привязка к CPU, карте памяти или паролю, 207
Программная карта, 144
Пустая трансферная карта, 149
Сравнение и синхронизация, 1399
Трансферная карта, 140
Утерянный пароль, 149
Проектное представление, 39, 40
Проекты для серийного оборудования (управление конфигурацией), 157
Промышленная среда
Разрешения, 1437
Просмотр MAC- и IP-адресов, 824
Просмотр топологии, 39
Простой линии, 1129
Протокол
ISO on TCP, 828
Modbus, 1120
PROFINET RT, 828
TCP, 828
UDP, 828
USS, 1120
Коммуникация, 1120
Свободно программируемая коммуникация, 1120
Протокол ISO-on-TCP, 828
Протокол UDP, 828
Протокол свободно программируемой коммуникации, 1120
Протоколы, коммуникация, 830
Псевдонимы на пользовательских веб-страницах, 1080
Пуск после СЕТЬ ВКЛ, 87
Обработка при запуске, 90
Пусковой ОВ, 92

P

Рабочая память, 28
CPU 1211C, 1444
CPU 1212C, 1457
CPU 1214C, 1470
CPU 1215C, 1484
CPU 1217C, 1499
Рабочая фаза
HSC (высокоскоростные счетчики), 576
Рабочее состояние, 46, 47
Переключение STOP/RUN, 1397
Рабочие состояния CPU, 87
Рабочее состояние RUN, 87, 90, 1397
Панель управления, 47
Функция принудительной установки, 1409
Экранные кнопки на панели инструментов, 46
Рабочее состояние STOP, 87, 1397
Панель управления, 47
Разрешение выходов в режиме STOP, 1407
Функция принудительной установки, 1409
Экранные кнопки на панели инструментов, 46
Радиосвязь с веб-сервером, 1031
Различия
в инструкциях Modbus RTU, 1267
в инструкциях Modbus TCP, 1198
в инструкциях PtP, 1120
в инструкциях TCON, TDISCON, TSEND и TRCV, 863
в инструкциях TSEND_C и TRCV_C, 842
в инструкциях USS, 1177
Различные ошибки параметров PtP, 1140
Разработка решения автоматизации с устройством PLC, 179, 180
Разрешение, сертификация в Корее, 1436
Разрешения
ATEX, 1435
CE, 1433
cULus, 1434
FM, 1434
Австралия и Новая Зеландия - RCM Mark, 1436
Морские допуски, 1436
Сертификация в Корее, 1436
Разъем RS485
Оконечная нагрузка, 1118
Расчетные напряжения, 1440
Расширение возможностей S7-1200, 31
Расширенный интерфейс блока
Загрузка в рабочем состоянии RUN, 1416
Расширительный кабель, 1628
Демонтаж, 68
Монтаж, 68
Расширяемые инструкции, 44
Регистрация значений Онлайн-DB, 1403
Регулируемое управление перемещением
PROFIdrive, 676
Аналоговый привод, 676
Конфигурирование оси, 676
Редактирование в рабочем состоянии RUN, (Загрузка в рабочем состоянии RUN)

Редактор текстов программы
Мониторинг, 1402
Состояние, 1402
Резервирование
Домены резервирования, 973
Клиенты резервирования, 971
Резервирование среды
Конфигурирование, 975
Принцип работы в кольцевой топологии, 972
Резервное копирование CPU, 1428
Рецептура
RecipeExport (экспорт рецептуры), 504
RecipeImport (импорт рецептур), 506
Обзор, 499
Пример программы, 508
Структура DB, 500
Руководства
Заземление, 76
Индуктивные нагрузки, 79
Ламповые нагрузки, 78
Монтаж CPU, 60
Порядок действий при монтаже, 59
Правила электромонтажа, 74, 76
Развязка, 75
Установка, 53
Руководство по изоляции, 75
Ручное управление фрагментным DB, 1108

C

Сброс значений DB, 1403
Сброс на заводские настройки, 1393
Сброс таймера (RT), 227
Сведения о программе
В структуре вызовов, 217
Светодиодная индикация
Коммуникационный интерфейс, 1117, 1387
Состояние CPU, 1387
Свойства CPU, пользовательские веб-страницы
Конфигурация STEP 7, 1087
Установка нескольких языков, 1108
Свойство синхронизации времени, 826
Сглаживающие блоки схемной защиты для индуктивных нагрузок, 79
Сервер OPC UA
Активация, 1017
Аутентификация пользователя, 1019
Доверенные клиенты, 1019
Интерфейсы, 1021
Настройки конфигурации, 1018
Обзор, 1017
Обзор вопросов безопасности, 1018
Определение интерфейса сервера, 1019
Поддерживаемые правила безопасности, 1018
Поддерживаемые типы данных, 1020
Поддерживаемые типы узлов, 1020
Спецификация компаньона, 1021
Узлы PLC, 1020
Сервисная и техническая поддержка, 3
Сериализация, 263
Сертификат безопасности Siemens, веб-страницы, 1042
Сертификация по ATEX, 1435
Сертификация по cULus, 1434
Сертификация по FM, 1434
Сетевая коммуникация, 931
Концевое сопротивление, 1118
Сетевое соединение
Несколько CPU, 935, 936, 940, 988, 992
Соединение устройств, 811
Сигнальные модули
SM 1222 DQ 16 x 24 VDC потребитель, 1521
Сигнальные модули (SM)
SM 1221 DI 16 x 24 VDC, 1517
SM 1221 DI 8 x 24 VDC, 1517
SM 1222 DQ 16 x 24 VDC, 1521
SM 1222 DQ 16 x Relay, 1521
SM 1222 DQ 8 x 24 VDC, 1519
SM 1222 DQ 8 x Relay, 1519
SM 1222 DQ 8 x Relay (переключение), 1519
SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC, 1527
SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC потребитель, 1527
SM 1223 DI 16 x 24 VDC, DQ 16 x Relay, 1527
SM 1223 DI 8 x 120/230 VAC/DQ 8 x Relay, 1534
SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC, 1527
SM 1223 DI 8 x 24 VDC, DQ 8 x Relay, 1527
SM 1231 AI 4 x 13 бит, 1537
SM 1231 AI 4 x 16 бит, 1537
SM 1231 AI 4 x 16 бит TC, 1550
SM 1231 AI 4 x RTD x 16 бит, 1557
SM 1231 AI 8 x 13 бит, 1537
SM 1231 AI 8 x 16 бит TC, 1550
SM 1231 AI 8 x RTD x 16 бит, 1557
SM 1232 AQ 2 x 14 бит, 1541
SM 1232 AQ 4 x 14 бит, 1541
SM 1234 AI 4 x 13 бит / AQ 2 x 14 бит, 1543
SM 1278 4xIO-Link-Master, 1563
Время реакции на скачок, 1547
Демонтаж, 65
Добавление модулей, 156
Конфигурация параметров, 174
Монтаж, 64
Обзор, 32

Потребляемая мощность, 1633
Представление аналогового входа (напряжение), 1548, 1594
Представление аналогового входа (ток), 1548, 1595
Представление аналогового выхода (напряжение), 1549, 1596
Представление аналогового выхода (ток), 1550, 1596
Расширительный кабель, 68
Сигнальные платы (SB)
SB 1221 DI 4 x 24 VDC, 200 кГц, 1577
SB 1221 DI 4 x 5 VDC, 200 кГц, 1577
SB 1222 DQ 4 x 24 VDC, 200 кГц, 1580
SB 1222 DQ 4 x 5 VDC, 200 кГц, 1580
SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC, 1587
SB 1223 DI 2 x 24 VDC/DQ 2 x 24 VDC, 200 кГц, 1583
SB 1223 DI 2 x 5 VDC/DQ 2 x 5 VDC, 200 кГц, 1583
SB 1231 AI 1 x 12 бит, 1590
SB 1231 AI 1 x 16 бит RTD, 1602
SB 1231 AI 1 x 16 бит Thermocouple, 1597
SB 1232 AQ 1 x 12 бит, 1592
Демонтаж, 62
Добавление модулей, 156
Конфигурация параметров, 174
Монтаж, 62
Обзор, 31
Потребляемая мощность, 1633
Представление аналогового выхода (напряжение), 1549, 1596
Представление аналогового выхода (ток), 1550, 1596
Представление входа (напряжение), 1548, 1594
Представление входа (ток), 1548, 1595
Символ для конца сообщения, 1133
Символ для начала сообщения, 1129
Синхронизация
Свойство синхронизации времени (PROFINET), 826
Синхронизация времени, 177
Синхронизация стартовых значений блоков данных, 212
Система ввода-вывода, обмен данными, 947
Системные требования, 37
Системные часы
RD_SYS_T (считывание реального времени), 346
WR_LOC_T (установка местного времени), 346
WR_SYS_T (установка реального времени), 346
Скорость выполнения, 1446, 1459, 1472, 1486, 1501
Скорость выполнения инструкций, 1446, 1459, 1472, 1486, 1501
Скорость передачи, 1124
Смартфон, доступ к веб-серверу, 1031
События безопасности в буфере диагностики, 114
Совместимость, 50
Согласование локальных данных в структуре вызова, 120
Соединение между локальным и партнерским CPU, 812
Соединения
Количество соединений (PROFINET/ PROFIBUS), 808
Соединения с несколькими узлами
Ethernet протоколы, 1006
Типы соединений, 1006
Создание DB для пользовательских веб-страниц, 1088
Создание пользовательских веб-страниц, 1071
Создание сетевого соединения
Между устройствами PLC, 811
Сообщение
Длина, 1133
Конец, 1131
Пуск, 1128
Состояние
Светодиодная индикация, 1387
Светодиодные индикаторы (коммуникационные интерфейсы), 1117
Сохранение резервных файлов, 1429
Сохраняемые переменные (теги) блока
Загрузка в рабочем состоянии RUN, 1417
Сохраняющая память, 28, 109
CPU 1211C, 1444
CPU 1212C, 1457
CPU 1214C, 1470
CPU 1215C, 1484
CPU 1217C, 1499
Специальные символы
Пользовательские веб-страницы, 1085
Справочники/руководства, 4
Сравнение значений, 242
Сравнение и синхронизация онлайн и офлайн CPU, 1399
Сравнительная таблица
Варианты CPU, 28
Модули, 31
Устройства HMI, 32
Срез (тегированного типа данных), 133

Срок службы карты памяти, 146
Срок службы реле, 1441
Стандартная DIN-рейка, 59
Стандартная вводная веб-страница, 1041
Стандартная веб-страница "Диагностика", 1043, 1045
Стандартная веб-страница "Информация о модулях", 1046
Стандартная веб-страница "Коммуникация", 1050
Стандартная веб-страница "Начальная страница", 1042
Стандартная веб-страница "Состояние переменных (тегов)", 1055
Стандартные блоки данных, 187
Стандартные веб-страницы, 1023
  JavaScript, 1113
  Вводная страница, 1041
  Входы в систему и выход из системы, 1039
  Диагностика, 1043, 1045
  Доступ через PC, 1029
  Защищенный доступ, 1031
  Изменение рабочего состояния, 1042
  Информация о модулях, 1046
  Коммуникация, 1050
  Начальная страница, 1042
  Ограничения для куки, 1113
  Состояние переменных (тегов), 1055
  Структура, 1036
Стандартные ведомые устройства DP
  Запись всех выходов с SETIO, 391
  Запись части выходов с помощью SETIO_PART, 394
  Считывание всех входов с помощью GETIO, 390
  Считывание части входов с помощью GETIO_PART, 392
Станция
  Чтение информации с помощью GetStationInfo, 458
Старая инструкция TRCV_C (получение данных через Ethernet (TCP)), 856
Старая инструкция TSEND_C (передача данных через Ethernet (TCP)), 856
Старые инструкции TCON, TDISCON, TSEND и TRCV, 876
Степень загрязнения, 1440
Степень загрязнения/категория перенапряжени, 1440
Степень защиты, 1440
  Утерянный пароль, 149
Столбцы и заголовки в окнах задач, 44
Стоповые биты, 1124
Структура вызовов, 217
Структурное программирование, 180
Схемы электрических соединений
  CB 1241 RS 485, 1620
  CPU 1211C, 1454
  CPU 1212C, 1467
  CPU 1214C, 1480
  CPU 1215C, 1495
  CPU 1217C, 1512
  SB 1221, 1579
  SB 1222, 1582
  SB 1223, 1586, 1589
  SB 1231, 1591
  SB 1231 RTD, 1604
  SB 1231 Thermocouple, 1601
  SB 1232, 1593
  SM 1221, 1518
  SM 1222, 1523
  SM 1223, 1531, 1536
  SM 1231, 1539
  SM 1231 RTD, 1559
  SM 1231 Thermocouple (TC), 1552
  SM 1232, 1542
  SM 1234, 1546
  SM 1278 IO-Link-Master, 1566
Счетчик
  CTD (обратный счёт), 236
  CTRL_HSC (управление высокоскоростным счетчиком), 587
  CTRL_HSC_EXT (управление высокоскоростным счетчиком (расширенное)), 558
  CTU (прямой счёт), 236
  CTUD (прямой и обратный счёт), 236
  Количество, 30
  Конфигурация HSC, 574
  Операция (стандартный счетчик), 237
  Размер, 30
Счетчик часов работы (RTM), 350
Счетчики
  Количество, 1447, 1460, 1473, 1487, 1502
  Размер, 1447, 1460, 1473, 1487, 1502

T

Таблица принудительно задаваемых значений
  Обращение к входам периферии, 1408
  Принудительное присваивание значений, 1408
  Функция принудительной установки, 1409
Таблица текущего контроля
  Запуск значений, 1406
  Мониторинг, 1401

Принудительное присваивание значений, 215
Принцип работы, 1404
Разрешение выходов в режиме STOP, 1407
Таймеры
RT (сброс таймера), 227
TOF (задержка выключения), 227
TON (задержка включения), 227
TONR (накопитель времени), 227
TP (импульс), 227
Количество, 30, 1447, 1460, 1473, 1487, 1502
Принцип работы, 230
Размер, 30, 1447, 1460, 1473, 1487, 1502
Телеуправление, 1371
Коммуникационные процессоры, 1367
Тепловая зона, 54, 58
Терминальный блок, 67, 1642
Терминальный блок, монтаж и демонтаж, 67
Термопара
SB 1231 AI 1 x 16 бит, 1597
Компенсация холодного спая, 1554, 1599
Общий принцип работы, 1554, 1599
Таблица выбора термопары для SM 1231, 1554
Таблица выбора фильтра SM 1231 Thermocouple (TC), 1554
Таблица выбора фильтра для SB 1231, 1601
Таблицы выбора фильтра для SB 1231 Thermocouple (TC), 1599
Тестирование в рабочем состоянии RUN, 1410, 1419
Тестирование программы, 215
Техническая поддержка, 3
Техническая поддержка Siemens, 3
Технические данные, 1433
BB 1297, 1608
CB 1241 RS485, 1620
CM 1241 RS232, 1621
CM 1241 RS422/485, 1622
CPU 1211C AC/DC/RLY, 1444
CPU 1211C DC/DC/DC, 1444
CPU 1211C DC/DC/RLY, 1444
CPU 1212C AC/DC/RLY, 1457
CPU 1212C DC/DC/DC, 1457
CPU 1212C DC/DC/RLY, 1457
CPU 1214C AC/DC/RLY, 1470
CPU 1214C DC/DC/DC, 1470
CPU 1214C DC/DC/RLY, 1470
CPU 1215C AC/DC/RLY, 1484
CPU 1215C DC/DC/DC, 1484
CPU 1215C DC/DC/RLY, 1484
CPU 1217C DC/DC/DC, 1499
SB 1221 DI 4 x 24 VDC, 200 кГц, 1577
SB 1221 DI 4 x 5 VDC, 200 кГц, 1577
SB 1222 DQ 4 x 24 VDC, 200 кГц, 1580
SB 1222 DQ 4 x 5 VDC, 200 кГц, 1580
SB 1223 DI 2 x 24 VDC, DQ 2 x 24 VDC, 1587
SB 1223 DI 2 x 24 VDC/DQ 2 x 24 VDC, 200 кГц, 1583
SB 1223 DI 2 x 5 VDC/DQ 2 x 5 VDC, 200 кГц, 1583
SB 1231 AI 1 x 12 бит, 1590
SB 1231 AI 1 x 16 бит RTD, 1602
SB 1231 AI 1 x 16 бит Thermocouple, 1597
SB 1232 AQ 1 x 12 бит, 1592
SM 1221 DI 16 x 24 VDC, 1517
SM 1221 DI 8 x 24 VDC, 1517
SM 1222 DQ 16 x 24 VDC, 1521
SM 1222 DQ 16 x 24 VDC потребитель, 1521
SM 1222 DQ 16 x Relay, 1521
SM 1222 DQ 8 x 24 VDC, 1519
SM 1222 DQ 8 x Relay, 1519
SM 1222 DQ 8 x Relay (переключение), 1519
SM 1223 DI 16 x 24 VDC, DQ 16 x 24 VDC, 1527
SM 1223 DI 16 x 24 VDC, DQ 16 x Relay, 1527
SM 1223 DI 8 24 VDC, DQ 8 x Relay, 1527
SM 1223 DI 8 x 120/230 VAC/DQ 8 x Relay, 1534
SM 1223 DI 8 x 24 VDC, DQ 8 x 24 VDC, 1527
SM 1231 AI 4 x 13 бит, 1537
SM 1231 AI 4 x 16 бит, 1537
SM 1231 AI 4 x 16 бит TC, 1550
SM 1231 AI 8 x 13 бит, 1537
SM 1231 AI 8 x 16 бит TC, 1550
SM 1232 AQ 2 x 14 бит, 1541
SM 1232 AQ 4 x 14 бит, 1541
SM 1234 AI 4 x 13 бит / AQ 2 x 14 бит, 1543
SM 1278 4xIO-Link-Master, 1563
Время реакции на скачок (CPU), 1453, 1466, 1479, 1493, 1510
Время реакции на скачок (SB), 1594
Время реакции на скачок (SM), 1547
Имитаторы входных сигналов, 1625
Карты памяти, 1624
Модуль потенциометров, 1627
Общие технические данные, 1433
Представление аналогового входа (напряжение), 1548, 1594
Представление аналогового входа (ток), 1548, 1595
Представление аналогового выхода (напряжение), 1549, 1596
Представление аналогового выхода (ток), 1550, 1596
Промышленная среда, 1437
Разрешения, 1433
Расчетные напряжения, 1440

Сигнальный модуль SM 1231 AI 4 x RTD x 16 бит, 1557
Сигнальный модуль SM 1231 AI 8 x RTD x 16 бит, 1557
Условия окружающей среды, 1439
Электромагнитная совместимость (ЭМС), 1437
Технологические инструкции, 558, 587
Технологические объекты
ПИД, 592
Управление перемещением, 665
Технологический модуль, SM 1278 4xIO-Link-Master, 1563
Типы данных, 123
Bool, Byte, Wort и Doppelwort, 125
Real, LReal (плавающая запятая), 126
Struc, 132
Time, Date, TOD (время), DTL (Date and Time long), 127
USInt, SInt, UInt, Int, UDInt, Dint (целое число), 126
Variant (указатель), 133
Массивы, 131
Редактор типов данных PLC, 132
Символы и строки, 129
Типы перечисления на пользовательских веб-страницах, 1081, 1082
Типы счета
Высокоскоростные счетчики, 575
Топология
Кольцо, 971
Трансферная карта, 140
Вставить в CPU, 137
Конфигурирование параметров запуска, 139
Обзор, 136
Пустая трансферная карта для утерянного пароля, 149
Утерянный пароль, 149
Трансферная карта (программная карта), 1624
Тригонометрические инструкции, 254

У

Указатель
Тип данных Variant, 133
Управление конфигурацией (управление опциями), 157
Набор управляющих данных, 160
Пример, 163
Управление опциями (управление конфигурацией), 157
Управление перемещением
MC_CommandTable, 756
MC_MoveAbsolute (абсолютное позиционирование оси), 746
Аппаратные и программные конечные выключатели, 716
Выполнение движения оси к референтной точке, 726
Движение к референтной точке (порядок действий при активном реферировании), 733
Инструкция MC_ChangeDynamic (изменение настроек динамики для оси), 759
Инструкция MC_Halt (приостановка оси), 744
Инструкция MC_Home (выполнение движения оси к референтной точке), 741
Инструкция MC_MoveJog (перемещение оси в толчковом режиме), 754
Инструкция MC_MoveRelative (относительное позиционирование оси), 749
Инструкция MC_MoveVelocity (перемещение оси с заданной скоростью), 751
Инструкция MC_Power (деблокировка/блокировка оси), 737
Инструкция MC_ReadParam (чтение параметров технологического объекта), 763
Инструкция MC_Reset (подтвердить ошибку), 740
Инструкция MC_WriteParam (запись параметров технологического объекта), 761
Обзор, 656
Параметры конфигурации для движения к референтной точке, 729
Список ErrorID (идентификаторы ошибок) и ErrorInfo (информация об ошибках), 777
Телеграмма 4, 686
Технологический модуль (TM) Pulse, 704
Фазирование, 662
Управление потоком, 1124
Конфигурация, 1124
Управлять, 1125
Управления программой (SCL), 325
CASE, 327
CONTINUE, 331
EXIT, 332
FOR, 329
GO TO, 333
IF-THEN, 326
REPEAT, 331
RETURN, 333
WHILE, 330
Управляющий DB для пользовательских веб-страниц
Глобальные команды, 1108

Команды и состояния запроса, 1108
Параметры WWW инструкции, 1089
Уровень защиты
CPU, 203
Блок кода, 206
Привязка к CPU, карте памяти или паролю, 207
Условия начала, 1128
Условия окончания, 1131
Условия окружающей среды
Транспортировка и хранение, 1439
Условия эксплуатации, 1439
Установка
TS адаптер SIM карта, 71
TS адаптер и TS модуль, 70
Баланс мощности, 55
Устойчивость к динамическим изменениям напряжения питания, 1438
Устранение ошибок
Буфер диагностики, 1397
Светодиодная индикация, 1387
Устройства HMI
Конфигурирование коммуникации PROFINET, 934
Обзор, 32
Сетевое соединение, 811
Устройства PROFINET IO
Запись всех выходов с SETIO, 391
Запись части выходов с помощью SETIO_PART, 394
Считывание части входов с помощью GETIO_PART, 392
Устройство
PROFINET IO, 940
Имя PROFINET IO-устройства, 941
общего доступа, 956
Устройство PLC
Использование блоков, 180
Устройство общего доступа
Конфигурация, 956
Концепция, 952
Утерянный пароль, 149

Ф

Фазирование, 662
Фазовый сдвиг, OB циклических прерываний, 93
Файл GSD, 952
Фиксированная длина, 1133
Форматирование карты памяти, 1396
Фрагментный DB (пользовательские веб-страницы)
Генерация, 1088
Импорт с помощью AWP-команды, 1084
Создание с помощью AWP-команд, 1083
Функции, новые, 35
Функциональность, I устройство, 943
Функциональный блок (FB)
Блок данных экземпляра, 185
Вызов блоков кода в программе пользователя, 182
Выходные параметры, 185
Защита ноу-хау, 206
Линейные и структурированные программы, 180
Начальное значение, 185
Обзор, 83, 185
Одиночный FB с мультиэкземплярными DB, 186
Правильные номера FB, 83
Функция (FC)
Вызов блоков кода в программе пользователя, 182
Защита ноу-хау, 206
Линейные и структурированные программы, 180
Обзор, 83, 184
Правильные номера FC, 83
Функция трассировки, 1421

Ц

Цикл
Обзор, 107
Функция принудительной установки, 1409
Цифровые I/O
Захват импульсов, 174
Индикация состояния, 1388
Конфигурация, 174
Цифровые сигнальные модули
SM 1221, 1517
SM 1222, 1519, 1521
SM 1223, 1527, 1534
Цифровые сигнальные платы
SB 1221, 1577
SB 1222, 1580
SB 1223, 1583, 1587

Ч

Часто задаваемые вопросы, 4
Частота, тактовые биты, 113

Часы
RD_LOC_T (считывание местного времени), 346
RD_SYS_T (считывание реального времени), 346
WR_LOC_T (установка местного времени), 346
WR_SYS_T (установка реального времени), 346
Часы реального времени, 114
Четность, 1124
Числа
Real, 126
Двоичные, 125
Целое число, 126
Чтение из DB, I/O или памяти, 199, 271
Чтение переменных HTTP, 1077

Ш

Шинный соединитель, 32

Э

Экранные кнопки RUN/STOP, 46
Электромагнитная совместимость (ЭМС), 1437, 1438
Эмулятор терминала в примере программы PtP, 1175

Я

Языки, пользовательские веб-страницы, 1105